

# **Plausibilität im Planungsprozess**

## **Umbau und Umnutzung als Optimierungsaufgabe**

### **Dissertation**

zur Erlangung des akademischen Grades

Doktor-Ingenieur

an der Fakultät Architektur

der

Bauhaus-Universität Weimar

vorgelegt von

Doz. M.Sc. Arch. Thorsten Michael Lömker

geb. am 13.08.1965

Weimar, 2006

Gutachter:

1. Prof. Dr.-Ing. Dirk Donath, Bauhaus-Universität Weimar
2. Prof. Dr.ès.sc.techn. Niklaus Kohler, Universität Karlsruhe (TH)
3. Prof. Dr.-Ing. Karl Beucke, Bauhaus-Universität Weimar

Tag der Disputation: 27.09.2006

## Danksagung

Mein erster Dank gilt einem ehemaligen Kollegen, Herrn Dipl.-Ing. Radomir Markovic. Meine ungeduldigen Fragen zur Computernutzung beantwortete er stets mit einem knappen: „Herr Lömker, lesen Sie!“

Mein besonderer Dank gilt natürlich meinem Mentor, Herrn Prof. Dr.-Ing. Dirk Donath, der diese Arbeit kritisch begleitet hat und mir die Möglichkeit gab, für 3 Jahre an seinem Lehrstuhl tätig zu sein. Herrn Prof. Dr.ès.sc.techn. Niklaus Kohler und Herrn Prof. Dr.-Ing. Karl Beucke danke ich herzlich für ihre Bereitschaft, meine Dissertationsschrift zu begutachten.

Frau Petzold danke ich sehr für die orthographische Durchsicht der vorliegenden Arbeit. Meiner Lebensgefährtin Katharina Richter für die unzähligen abendlichen Diskussionen über Architektur und Forschung.

Mein größter Dank aber gilt meinen Eltern, die meine Wege und Ziele immer unterstützten.

# Thesen

## Grundlagen

### These 1

Der ungenutzte Gebäudebestand in der Bundesrepublik Deutschland deckt zum größten Teil den Bedarf an Wohn- und Arbeitsraum. Deshalb liegt die planerische Herausforderung im Umbau und in der Umnutzung dieser Gebäude.

### These 2

Umbau und Umnutzung sind Strategien zur Erhaltung existenter Werte. Aus diesem Grund müssen ungenutzte Bestandsgebäude einer Weiternutzung zugeführt werden.

### These 3

Das Ziel jeder Revitalisierungsmaßnahme ist die ressourcenschonende Weiternutzung des Gebäudebestandes. Deshalb muss die Entscheidung, ob und wie ein Gebäude revitalisiert wird, nicht nur unter ökonomischen, sondern auch unter ökologischen Gesichtspunkten getroffen werden.

### These 4

Die Entscheidung, ob ein Gebäude umgebaut oder umgenutzt werden kann, wird zu Beginn der Planungstätigkeit getroffen. Deshalb muss die Durchführbarkeit einer Revitalisierungsmaßnahme zu diesem Zeitpunkt plausibel dargelegt werden. Es gibt keine computergestützten Werkzeuge, die den Architekten bei der Lösung dieser Aufgabe unterstützen.

## **Revitalisierung - Potential**

### **These 5**

Der Gebäudebestand ist heterogen und nicht eindeutig typologisierbar. Dennoch kann ein Merkmal aus dem Bestand extrahiert werden, an Hand dessen nahezu jedes Gebäude kategorisiert werden kann. Dieses Merkmal ist die Konstruktion des Gebäudes.

### **These 6**

Fast jedes komplexe Gebäude kann entweder als Gebäude mit massiver Konstruktion oder als Gebäude mit leichter Konstruktion kategorisiert werden.

### **These 7**

Gebäude mit massiven Konstruktionen sind gekennzeichnet durch die schwierige Trennbarkeit von Primär-, Sekundär- und Tertiärstruktur. Ihre Grundrissorganisation ist demzufolge unflexibel und kann nur mit erheblichen Eingriffen in die Bausubstanz in größerem Umfang verändert werden.

### **These 8**

Gebäude mit leichten Konstruktionen sind gekennzeichnet durch die einfache Trennbarkeit von Primär-, Sekundär- und Tertiärstruktur. Ihre Grundrissorganisation ist demzufolge flexibel und kann mit geringen Eingriffen in die Bausubstanz in größerem Umfang verändert werden.

## **Revitalisierung - Strategien**

### **These 9**

Der Umbau eines Gebäudes ist mit dem Abbruch von Bauteilen verbunden. Daher kann der Umbau als zerstörungsreiche Vorgehensweise bezeichnet werden.

### **These 10**

Die Umnutzung eines Gebäudes kann ohne den Abbruch von Bauteilen erfolgen. Daher kann die Umnutzung als zerstörungsarme Vorgehensweise bezeichnet werden.

### **These 11**

Unter der Bedingung des nachhaltigen Umgangs mit dem Gebäudebestand ist die Weiternutzung eines Gebäudes unter größtmöglicher Beibehaltung der existenten Bausubstanz und unter geringstmöglicher Veränderung des baulichen Zustands zu erreichen. Die Umnutzung eines Gebäudes kann daher als die ökologischste Revitalisierungsmaßnahme bezeichnet werden. Vor der Entscheidung für einen Umbau ist demzufolge die Durchführbarkeit einer Umnutzung oder einer Kombination aus Umnutzung und Umbau zu prüfen.

### **These 12**

Im unter ökologischen Gesichtspunkten betrachteten Idealfall des Verzichts auf bauliche Eingriffe ist es ausreichend, ein Gebäude organisatorisch zu verändern, so dass durch den Tausch von Flächen auf neue Nutzerprofile abgestimmte veränderte Nutzungseinheiten gebildet werden können.

## **Revitalisierung - Methodik**

### **These 13**

Architekten verwenden unterschiedliche Methoden zur Lösung von Entwurfsproblemen. Aus diesem Grund gibt es keine allgemeingültige Vorgehensweise, die als Anhaltspunkt für die Entwicklung computergestützter Werkzeuge dienen könnte.

### **These 14**

Der Planungsprozess erfordert die Entwicklung von alternativen Lösungsmöglichkeiten einer Entwurfsaufgabe nach gleichen oder grundsätzlich verschiedenen Anforderungen. Diese Forderung kann im Fall einer Revitalisierungsmaßnahme derart komplex sein, dass ein Architekt sie nicht erfüllen kann.

### **These 15**

Das Entwerfen ist nicht frei von normativen Einflüssen. Gesetze, Verordnungen und Richtlinien müssen ebenso beachtet werden, wie gestalterische, wirtschaftliche oder technische Faktoren. Deshalb müssen normative Einflüsse als zu erfüllende Bedingungen oder Ziele betrachtet werden.

### **These 16**

Ein wesentliches Kriterium für oder gegen die Entscheidung, ein Gebäude zu revitalisieren, ist die Möglichkeit, den Ist-Zustand (Bestandsgrundriss) an den Soll-Zustand (Raumprogramm) anzupassen. Diese Anpassung stellt eine komplexe kombinatorische Problemstellung dar, deren potentieller Lösungsraum so groß ist, dass eine manuell entwickelte Lösung hinsichtlich der Existenz alternativer Lösungsmöglichkeiten nicht objektiv bewertet werden kann.

## **Modellbildung**

### **These 17**

Die zerstörungsreiche Vorgehensweise beim Umbau besitzt Analogien zur Methodik der Organisation von Warenlagern oder Containerterminals, bei denen Artefakte in Form von Waren durch Sortieren, Umschichten oder Verlagern dergestalt in ihrer Position verändert werden, dass eine Zielvorgabe („Gute Erreichbarkeit“) erfüllt wird.

### **These 18**

Die zerstörungsarme Vorgehensweise bei der Umnutzung besitzt Analogien zur Methodik von Brettspielen, wie Schach, Go oder 4Gewinnt, bei denen Artefakte in Form von Feldern oder Kreuzungspunkten von Rasterlinien dergestalt mit Spielsteinen belegt werden, dass eine Zielvorgabe („Spielgewinn“) erfüllt wird.

### **These 19**

Aus der Methodik der zitierten Analogien können für die zerstörungsreiche und die zerstörungsarme Vorgehensweise Modelle abgeleitet werden, die mit Hilfe der Informationstechnologie darstellbar sind.

## **Informationstechnologie**

### **These 20**

Umbau und Umnutzung sind kombinatorische Problemstellungen. Diese lassen sich als Optimierungsprobleme darstellen und mit Methoden des Operations Research lösen.

## **Systemkonzeption**

### **These 21**

Normative Einflüsse können in Optimierungsmodellen als Bedingungen definiert werden. Diese Bedingungen schränken den potentiellen Lösungsraum eines Revitalisierungsproblems ein, so dass untaugliche Lösungen vermieden werden.

### **These 22**

Entwurfsziele können in Optimierungsmodellen als Zielfunktionen definiert werden. Diese Zielfunktionen können gewichtet werden, so dass im Sinne der ressourcenschonenden Nutzung des Bestandes auch die nicht vollständige Erfüllung aller Ziele als Entscheidung für die Durchführung einer Revitalisierungsmaßnahme in Betracht gezogen werden kann.

### **These 23**

Optimierungsmodelle können die Anzahl der Lösungen für ein Revitalisierungsproblem umfassend erhöhen.

### **These 24**

In kürzerer Zeit können mit einem Optimierungsmodell mehr Lösungen eines Revitalisierungsproblems gefunden werden, als mit traditionellen Methoden.

### **These 25**

Die Performance einer mit einem Optimierungsmodell ermittelten Lösung eines Revitalisierungsproblems ist auf der Grundlage der vom Architekten aufgestellten Anforderungen objektiv messbar.

### **These 26**

Die tatsächliche Leistungsfähigkeit des Einsatzes von Optimierungsverfahren in der Revitalisierung kommt erst dann zum Tragen, wenn Umbau und Umnutzung als gleichberechtigte Alternativen zum Neubau in Betracht gezogen werden. Deshalb muss sich die Herangehensweise des Architekten an den Planungsprozess grundlegend verändern.



# Inhaltsverzeichnis

<b>Danksagung</b> .....	<b>2</b>
<b>Thesen</b> .....	<b>3</b>
<b>Inhaltsverzeichnis</b> .....	<b>9</b>
<b>1 Grundlagen</b> .....	<b>13</b>
1.1 Einführung .....	13
1.2 Motivation .....	14
1.3 Zielsetzung.....	16
1.4 Überblick .....	17
<b>2 Revitalisierung - Potential, Strategien, Methodik</b> .....	<b>19</b>
2.1 Revitalisierungspotential .....	19
2.1.1 Ausgangssituation .....	19
2.1.2 Historische Entwicklung .....	20
2.1.3 Kategorisierung des Gebäudebestandes .....	23
2.2 Revitalisierungsstrategien.....	26
2.2.1 Planungsgrundsätze .....	27
2.2.2 Revitalisierungsstrategien .....	28
2.2.2.1 Strategie 1 - Umbau .....	29
2.2.2.2 Strategie 2 - Umnutzung (Flächen- und Gebäudetausch) .....	33
2.2.2.3 Strategie 3 - Bauverzicht .....	34
2.3 Revitalisierungsmethodik .....	36
2.3.1 Planen und Entwerfen .....	36
2.4 Zusammenfassung .....	41
<b>3 Modellbildung</b> .....	<b>44</b>
3.1 Revitalisierungsmodelle .....	45
3.2 Zerstörungsreiches Modell .....	49
3.2.1 Methodik .....	49
3.2.2 Interdisziplinäre Analogien der Methodik .....	54
3.3 Zerstörungsarmes Modell .....	56
3.3.1 Methodik .....	56
3.3.2 Interdisziplinäre Analogien der Methodik .....	60
3.4 Zusammenfassung und Implikationen .....	60
3.4.1 Regeln.....	61
3.4.2 Mechanismen zur Problemlösung .....	63

<b>4</b>	<b>Informationstechnologie – Potential.....</b>	<b>65</b>
4.1	Wissensbasierte Systeme .....	65
4.1.1	Grundlagen .....	65
4.1.2	Komponenten Wissensbasierter Systeme .....	67
4.1.2.1	Erhebung von Wissen - Akquisitionskomponente .....	68
4.1.2.2	Speicherung von Wissen - Wissensbasis .....	69
4.1.2.3	Verarbeitung von Wissen - Inferenzkomponente .....	71
4.1.2.4	Interaktion mit dem Benutzer - Dialogkomponente.....	72
4.1.2.5	Darstellung der Ergebnisse - Erklärungskomponente .....	72
4.2	Wissensbasierte Systeme in der Architektur.....	73
4.2.1	Planungspraxis.....	73
4.2.2	Stand der Forschung und eigene Vorarbeiten .....	74
4.3	Bewertung aus architektonischer Sicht – Bescheidenheit .....	74
4.4	Zusammenfassung und Schlussfolgerung.....	75
<b>5</b>	<b>Systemkonzeption .....</b>	<b>80</b>
5.1	Eingabemodell .....	81
5.1.1	Beziehungen .....	81
5.2	Geometriemodell .....	83
5.2.1	Komponenten des Geometriemodells.....	83
5.3	Optimierungsmodell .....	88
5.3.1	Standardform eines Optimierungsmodells .....	89
5.3.2	Zerstörungsreiches Modell .....	93
5.3.2.1	Konstanten .....	94
5.3.2.2	Variablen .....	94
5.3.2.3	Nebenbedingungen.....	94
5.3.2.4	Obere und Untere Schranken .....	103
5.3.2.5	Zielfunktionen.....	103
5.3.2.6	Suchstrategien .....	104
5.3.3	Zerstörungsarmes Modell.....	105
5.3.3.1	Konstanten .....	106
5.3.3.2	Variablen .....	106
5.3.3.3	Nebenbedingungen.....	108
5.3.3.4	Obere und Untere Schranken .....	108
5.3.3.5	Zielfunktionen.....	108
5.3.3.6	Suchstrategien .....	109
5.4	Performancemodell.....	109

<b>6</b>	<b>Mathematische Modellbildung</b>	<b>110</b>
6.1	Mathematisches Modell 1 (Zerstörungsreiches Modell)	111
6.1.1	Ausgangsmodell	111
6.1.2	Erweiterung (1) des Ausgangsmodells	118
6.1.3	Erweiterung (2) des Ausgangsmodells	123
6.2	Mathematisches Modell 2 (Zerstörungsarmes Modell)	129
6.2.1	Ausgangsmodell	129
<b>7</b>	<b>Prototypen</b>	<b>133</b>
7.1	CATIA	134
7.1.1	Konzeptioneller Aufbau	134
7.1.2	Besonderheiten	135
7.1.3	Exemplarische Ergebnisse	136
7.1.4	Bewertung	137
7.2	VisualDOC	138
7.2.1	Konzeptioneller Aufbau	138
7.2.2	Besonderheiten	139
7.2.3	Exemplarische Ergebnisse	140
7.2.4	Bewertung	166
7.3	OPL	170
7.3.1	Konzeptioneller Aufbau	170
7.3.2	Besonderheiten	171
7.3.3	Exemplarische Ergebnisse (Zerstörungsreiches Modell)	174
7.3.3.1	Rechteckige Grundrisse und Units	174
7.3.3.2	Rechtwinklige Grundrisse und Units	199
7.3.4	Exemplarische Ergebnisse (Zerstörungsarmes Modell)	217
7.3.4.1	Rechtwinklige und Nicht-Rechtwinklige Grundrisse und Units	217
7.3.5	Bewertung	244
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>247</b>
8.1	Zusammenfassung	247
8.2	Bewertung	249
8.3	Ausblick	253
8.4	Schlussbetrachtung	256
	<b>Appendix A</b>	<b>257</b>
	A1 – Abkürzungen und Akronyme	258
	A2 – Abbildungen	260
	A3 – Tabellen	264

<b>Appendix B</b> .....	<b>266</b>
B1 – Mathematische Nomenklatur .....	267
B2 – Exemplarischer Programmcode .....	272
<b>Appendix C</b> .....	<b>286</b>
C1 – Curriculum Vitae .....	287
C2 – Veröffentlichungen .....	288
C3 – Copyright.....	291
C4 – Ehrenwörtliche Erklärung .....	293
<b>Literaturverzeichnis</b> .....	<b>294</b>
<b>Glossar</b> .....	<b>301</b>

# 1 Grundlagen

## 1.1 Einführung

Die Erneuerung, Umnutzung und Revitalisierung vorhandener Bausubstanz ist ein Prozess, der in der Geschichte des privaten und öffentlichen Gebäudebestandes schon immer zu beobachten war. Umbauten haben zu jeder Zeit und in allen gesellschaftlichen und kulturellen Kreisen stattgefunden. Die Gründe, die zu Umbauten führten, waren vielfältig. Veraltete Bausubstanz, veränderte Nutzungsanforderungen oder auch der Wandel ästhetischer Ansprüche und Grundpositionen veranlassten die Besitzer der Gebäude dazu, Umgestaltungen vorzunehmen. Auch in der heutigen Zeit haben sich die Faktoren, die zu Umnutzungen führten, kaum verändert. Es sind jedoch neue Faktoren hinzugekommen, die die Nutzung existenter Bausubstanz unter geänderten Rahmenbedingungen notwendig machen. Ökonomische und ökologische Größen spielen heute mehr denn je eine gewichtige Rolle bei der Instandsetzung und Revitalisierung von Gebäuden. Die nachhaltige Nutzung unserer gebauten Umwelt ist zu einem wichtigen Themenbereich in der Gesellschaft und der Architektur geworden. Doch auch wirtschaftliche Interessen prägen den Umgang und die Auseinandersetzung mit den baulichen Erzeugnissen der Vergangenheit. Somit werden vielfältige Anforderungen an diejenigen gestellt, die die Erneuerung und Revitalisierung der Bausubstanz zu verantworten und planerisch zu begleiten haben. Revitalisierungsmaßnahmen beschränken sich dabei nicht auf die bloße Instandsetzung des Gebäudebestandes. Vielmehr geht es bei den einzuleitenden Schritten um die Wiederbelebung von Gebäuden, die den heutigen Ansprüchen aus vielfältigen Gründen nicht mehr gerecht werden. Die Durchführung erfolgreicher Revitalisierungsmaßnahmen bedarf nicht nur konstruktiver, sondern auch wirtschaftlicher und ökonomischer Kenntnisse. Der Architekt ist in zunehmendem Maße gefordert, fachübergreifend zu denken und zu handeln und in Kooperation mit Spezialisten eine Gesamtlösung herbeizuführen. Diese interdisziplinäre Tätigkeit verlangt Vorgehensweisen und Fertigkeiten, die nicht selten zu Unsicherheiten im Planungsprozess führen. Im Gegensatz zur

Planung von Neubauvorhaben, sind die Rahmenbedingungen einer Revitalisierung oftmals ungewiss und schwierig zu kalkulieren. Die erfolgreiche Planung setzt jedoch genaue Kenntnisse und eine exakte Definition der Zielvorstellungen und Vorgehensweisen voraus. Es gibt eine Reihe von Faktoren, die das Planen, aber nicht notwendigerweise das Bauen im Bestand, rechtfertigen. Einer der wesentlichen Ansätze ist die Diskussion um die Nachhaltigkeit der Gebäudesubstanz. So ist das Umnutzen und Umbauen von Bestandsimmobilien unumstritten eine der ökologischsten und häufig auch ökonomischsten Vorgehensweisen des Bauens. „... die wesentlichen Steuergrößen werden nicht in den wenigen Neubauprojekten liegen, sondern in einer klugen Politik des Umgangs mit dem vorhandenen Baubestand: im Umbau, in Werterhaltungsstrategien, und im ‚Nichtbau‘.“ [Hassler und Kohler, 1998, S. 3]

## 1.2 Motivation

### Der Bogen und die Lyra

Architekten sind gerne Individualisten, doch wird ihnen alltäglich die Rolle der Generalisten und das Fachwissen der Spezialisten abverlangt. Während einerseits die Diskussion um den Architekten als Masterbuilder wieder auflebt [Glymph, Shelden et al., 2002], gibt es andererseits eine deutlich spürbare Negierung der Ambiguität des Architekturschaffens zu Gunsten der Individualität und der künstlerischen Aspekte des Berufes. Auch die Auseinandersetzung mit einer so unbeliebten Thematik wie der Revitalisierung, die gar den Verzicht auf das Neubauen proklamiert, oder die Divergenzen von Architektur und Computereinsatz zeigen, wie allgegenwärtig das Schwanken zwischen den Gegensätzlichkeiten in der Architektur ist.

Ambiguitäten finden sich auch in vielen anderen Bereichen. William Pedersen rezipiert eine derartige Doppeldeutigkeit im traditionellen japanischen Bogenschießen, die es einerseits als Sport und andererseits als geistige Tätigkeit erfahrbar macht [Pedersen, 1993]. Auch der Architektur wohnt eine solche relative Position inne. Sie erscheint als Wissenschaft, Kunst oder Technik, in einigen wenigen Fällen gar als kaufmännisches Handeln. Nach Pedersen repräsentieren zwei Symbole der griechischen Mythologie die Ambiguität der Architektur – der Bogen und die Lyra. Der Bogen ist ein System sich im Konflikt befindender Kräfte. Sein funktioneller Aufbau gleicht dem der Lyra. Wenn jedoch die divergierenden Kräfte, resultierend aus der Saite und dem Schaft des Bogens, mit der Saite der Lyra verglichen werden, erkennt man, dass aus den im Konflikt stehenden Kräften die schönste und vollkommenste Harmonie in Form vollendeter Töne entstehen kann. Für die Griechen repräsentierten der Bogen und die Lyra daher den Ausdruck eines der zentralen Aspekte des Lebens: Dualität. Diese Dualität stand für sie hinter allen Formen der Existenz und drückte sich aus in Gut und Böse, Hell und Dunkel, Geburt und Tod. Dabei verkörperten der Konflikt und die Harmonie alle denkbaren Formen der Dualität. Der Bogen beinhaltet von jeher eine ästhetische Einfachheit, die aus seiner inneren Komplexität erwächst. Seine Eleganz wird durch das Zusammenspiel zweier dominierender Komponenten, die in sich selbst gegensätzlich sind, erreicht. Die Sehne ist linear, von einfacher geometrischer Beschaffenheit und existiert unter Zug. Der Schaft hat eine komplexe Struktur mit wechselnden Querschnittsformen und steht unter Druck. Keines der beiden dominierenden Teile gewinnt die

Oberhand über das andere. Während des Spannens und des Lösens des Bogens spielen beide fluktuierende Rollen. Wechselseitig sind die zwei Komponenten entweder dominierend oder der anderen untergeordnet. Nie wird ein Zustand erreicht, in welchem auch nur eine der Komponenten vernachlässigbar wäre. Es ist ein System gleichgewichtiger Kräfte und Zustände, welches erreicht wird, durch das Zusammenspiel von Statik und Dynamik, Linearität und Krümmung, Einfachheit und Komplexität. Es ist die perfekte Symbolisierung jener Dualität, die – aus dem Wechselspiel von Konflikt und Harmonie – Ausgewogenheit entstehen lässt. Nur die Existenz der Kontradiktion führt zu vollkommener Perfektion.

Der Entstehungsprozess von Architektur unterliegt ebensolchen Kontradiktionen. Wenn Architektur den Prinzipien des Marcus Vitruvius Pollio folgt, dann ist der Architekt Künstler und Ingenieur und beides erscheint dann notwendig, gar unabdingbar, um einem Bauwerk Schönheit, Stabilität und Nützlichkeit zuteil werden zu lassen [Vitruvius, 1960]. Die Vereinigung von Dualität und Vielfalt der Architektur in einem Werk ist es, was den Architekten in seiner ihm ursprünglich obliegenden Position als Generalisten auszeichnet. Deshalb gehören sowohl künstlerische, als auch technische und kaufmännische Aspekte zu seinem Aufgabengebiet. Auch diese stehen nur scheinbar im Widerspruch zueinander, denn sie gewährleisten in Form „höchstmöglicher Vereinheitlichung größtmöglicher Diversifikation“ [Pedersen, 1993, S. 34] ein stimmiges und zweckmäßiges Bauwerk. Folgt man den Ausführungen Vitruvs wird deutlich, dass diese Grundprinzipien bereits in der Antike bekannt waren und als Notwendigkeit erkannt worden sind. So beschreibt Vitruv in den „Zehn Büchern über Architektur“ die Grundprinzipien dessen, was uns heute in der Architektur oftmals schwer zu schaffen macht. Im ersten Kapitel des ersten Buches verlangt er vom Architekten universale Bildung. Denn wie sonst könne ein Architekt die Anforderungen des Bauens und Planens so umsetzen, dass sie den Erfordernissen der Praxis gerecht werden. Vitruv zeigt gleichermaßen auf, dass sich das Bild des Baumeisters keineswegs auf Bauwerke im Sinne unseres heutigen Verständnisses begrenzen darf. So handeln Buch 9 und 10 von der Konstruktion von Uhren, Maschinen und Instrumenten [Eversberg, 1995]. Auch wenn die Beschäftigung mit derlei Technologien aus heutiger Sicht weit hergeholt erscheinen mag, macht Vitruv doch unmissverständlich klar, dass die Auseinandersetzung mit fachfremden Disziplinen fruchtbar für die planerische Tätigkeit sein kann. Die Präzision und Fertigkeit die verlangt wird, um mechanische Konstruktionen herzustellen und die Anwendung von Konstruktionsregeln, ergo die Formalisierung, vorausschauende Planung und Festlegung der Entwicklung eines Produktes, ist auch in der Revitalisierung von Bauwerken ein wesentlicher Bestandteil der Tätigkeit des Architekten. Das Verständnis für die Zusammensetzung eines existenten Bauwerkes zu entwickeln, seine Eigenheiten zu erkennen und die ihm zu Grunde liegenden Mechanismen zu erfassen, ist unabdingbar für die erfolgreiche Durchführung einer Revitalisierungsmaßnahme. Und auch für diese Mechanismen findet Vitruv in ‚Ordinatio‘ geometrische Bestimmungen, die es ihm ermöglichen, einem Bauwerk elementare Einheiten (modulus) zu Grunde zu legen, aus welchen das spätere Gebäude zusammengesetzt wird. Da diese Bauvorschriften, basierend auf dem ‚modulus‘, zahlreiche Verhältniszahlen der einzelnen Bauteile untereinander festlegen, ist die Bestimmung dieser Maßeinheit das zentrale Element am Anfang der Entwurfsarbeit. In ‚Dispositio‘ geht Vitruv sogar so weit, dass er für den Tempelbau nur bestimmte Ausprägungen von Säulen zulässt, die durch die Definition von Größenverhältnissen der Bauteile untereinander das spätere Bauwerk fast vollständig festlegen. Da Vitruv das „umgreifende Stümpertum“ beklagt, liegt die Vermutung nahe, dass er durch die Festlegung dieser

Entwurfsvorschriften erreichen wollte, dass auch weniger begabte Baumeister ihre Arbeit zu akzeptablen Ergebnissen führen können. Möglicherweise hat bereits Vitruv das Aufstellen von Regeln als Maßnahme zur Qualitätssicherung gesehen. Ohne jeden Zweifel benutzte er Regelwerke und griff bei der Planung und Ausführung seiner Bauten auf Kenntnisse anderer Fachdisziplinen zurück.

Die unzähligen Regeln, denen sich der Architekt in seiner Planungstätigkeit heute gegenübergestellt sieht, kann er, insbesondere in seiner Eigenschaft als Generalist, kaum bewältigen. Dem „Stümpertum“ des Vitruv steht also heute die Existenz des Architekten als Generalist gegenüber. Die Integration von Spezialisten in den Planungsprozess scheint von Nöten und wird in den einschlägigen Richtlinien und Verordnungen sogar gefordert [Hesse und Korbion, 1992]. Nun zeigt aber die Erfahrung, dass sich die Integration von Fachplanern zu Beginn des Planungsprozesses mitunter als beschwerlich herausstellt. Der Bauherr ist in der Regel erst dann gewillt, in die Beauftragung von Fachplanern zu investieren, wenn die Machbarkeit der Revitalisierungsmaßnahme vom Architekten nachgewiesen worden ist. Das bringt den Architekten letztlich in die Bedrängnis, schon im Vorfeld der Planung einen potentiellen Auftrag entweder zu verlieren oder eigenständig, mit allen damit verbundenen haftungsrechtlichen Konsequenzen, jene Fachplanungstätigkeiten zu verrichten, die für die Beurteilung der potentiellen Revitalisierungsmaßnahme notwendig sind. Zur Aufhebung dieser Kontradiktion ist es erforderlich Hilfsmittel zu entwickeln, die den Architekten in genau diesen Bereichen unterstützen, so dass zu Beginn des Planungsprozesses Entscheidungen getroffen werden können, die eine erfolgreiche Bearbeitung der Gesamtplanung zulassen. Mit sinngemäßer Selbstverständlichkeit, mit welcher im traditionellen japanischen Bogenschiessen die Frage nach geistiger Tätigkeit oder Sport nie gestellt wird, müssen dem Architekten Werkzeuge zur Verfügung gestellt werden, deren Aufgabe es ist, ihn bei der Bearbeitung der Planungsaufgabe inhaltlich zu unterstützen und eben jenes problem-spezifische Wissen zu ergänzen, das zu plausiblen Lösungen im Planungsprozess führt.

### 1.3 Zielsetzung

Die Bauaufgaben der Zukunft liegen in der Auseinandersetzung mit bestehender Architektur. Die planerische Herausforderung besteht im Verzicht auf den Neubau durch die Umnutzung und den Umbau existenter Gebäude.

Umnutzung und Umbau sind Werterhaltungsstrategien, die den Lebenszyklus eines Gebäudes als integralen Bestandteil der Planung betrachten und deren Ziel es ist, ungenutzte Bestandsgebäude durch keine oder wenige bauliche Eingriffe so zu verändern, dass sie einer Weiternutzung zugeführt werden können. Die Umnutzung unterliegt der Prämisse, dass an den Gebäuden keine baulichen Veränderungen vorgenommen werden, wohingegen der Umbau bauliche Eingriffe gestattet. Als Alternative zum Neubau ist der Erfolg beider Strategien entscheidend davon abhängig, dass der Architekt schon zu Beginn der Planung zu der Entscheidung gelangt, ob sich ein Gebäude unter Anwendung einer der beiden Strategien weiternutzen lässt. Diese Entscheidung wird vom Architekten in der Praxis durch einen Vergleich des Soll-Zustands (Raumprogramm) mit dem Ist-Zustand des Gebäudes getroffen. Die Analyse und Bewertung des Bestandes erfolgt in dieser frühen Phase der Planung in Form von Vorentwurfsskizzen, welche die



organisatorischen oder baulichen Veränderungen der Gebäudegrundrisse im Falle einer Weiternutzung darstellen.

Das wesentliche Entscheidungskriterium für die Revitalisierung eines Gebäudes ist die Erfüllung des Raumprogramms. Diese Vorgehensweise führt schnell zu Revitalisierungslösungen, die das Gebäude unter erheblichen baulichen Eingriffen in Richtung der Erfüllbarkeit des Raumprogramms verändern. Unter der Prämisse des nachhaltigen Umgangs mit dem Gebäudebestand ist die Entscheidungsfindung zur Weiternutzung eines Gebäudes jedoch unter größtmöglicher Beibehaltung der existenten Bausubstanz und unter geringstmöglicher Veränderung des baulichen Zustands zu erreichen. Diese Vorgehensweise findet bei Revitalisierungsmaßnahmen bislang wenig Beachtung.

In dieser Arbeit wird die Hypothese aufgestellt, dass der Vergleich des Raumprogramms mit dem Gebäudegrundriss im Wesentlichen eine kombinatorische Problemstellung darstellt. Unter dieser Annahme wird untersucht, ob durch den Einsatz von Optimierungsverfahren in der Grundrissplanung Lösungen für Umbau- und Umnutzungsaufgaben automatisiert erzeugt werden können. Diese Lösungen sollen durch die Neuordnung oder den Tausch existenter Flächen erzeugt werden. Ziel ist es, durch den computergestützten Einsatz dieser Verfahren zu plausiblen Planungslösungen, die dem Architekten als Grundlage für die weitere Bearbeitung der Planung dienen, zu gelangen.

## 1.4 Überblick

Kapitel 1 gibt eine kurze Einführung in die Problematik des Bauens im Bestand. Es beschreibt die schwierigen Voraussetzungen, unter denen Architekten sich mit dem Neubau und der Revitalisierung, der Architektur und dem Computer auseinandersetzen müssen, und leitet daraus die Motivation für diese Arbeit ab. Es wird exemplarisch aufgezeigt, weshalb die Auseinandersetzung mit der Thematik des Bauens im Bestand notwendig ist und welche informationstechnischen Werkzeuge für die Unterstützung der Grundrissplanung im Bestand untersucht werden.

Ein historischer Überblick in Kapitel 2 verdeutlicht, welche Bestandsbauten gegenwärtig zur Revitalisierung anstehen und welchen Einfluss der Zeitgeist, das politische Geschehen und die Architekturtheorien auf die Konzeption von Vorgehensweisen für die Revitalisierung von Gebäuden haben. Es werden drei Revitalisierungsstrategien (Umbau, Umnutzung, Bauverzicht) vorgestellt, die als Grundlage der Entwicklung von Werkzeugen für die Bearbeitung von Revitalisierungsaufgaben verwendet werden können. Das Kapitel schließt mit einer Analyse der methodischen Vorgehensweise des Architekten im Planungsprozess. Es wird diskutiert, welche Relevanz die zur Anwendung gelangenden Planungsmethoden für die Entwicklung von Entwurfssystemen haben.

In Kapitel 3 werden auf der Grundlage der Umbau- und Umnutzungsstrategien zwei Modelle gebildet, unter deren Anwendung zerstörungsreiche und zerstörungsarme Revitalisierungsmaßnahmen durchgeführt werden können. Die Modelle sind unabhängig von der Verwendung von Computern, stellen jedoch eine formalisierte Vorgehensweise dar.

Das Potential der Informationstechnik für die Umsetzung des Zerstörungsreichen und Zerstörungsarmen Modells wird in Kapitel 4 herausgearbeitet. Es werden Wissensbasierte Systeme beschrieben und hinsichtlich ihrer Anwendbarkeit auf architektonische Fragestellungen untersucht. Es wird verdeutlicht, weshalb die Verwendung von Wissen in Form von Regelwerken besondere Bedeutung für die Revitalisierung von Bestandsbauten besitzt.

Kapitel 5 befasst sich mit der Frage, wie derartige Regeln beschrieben und verarbeitet werden können. Es wird eine Systemkonzeption für die informationstechnische Darstellung des Zerstörungsreichen und des Zerstörungsarmen Modells entwickelt, mit dem Ziel, Umbau- und Umnutzungsaufgaben durch die Verwendung von Optimierungsverfahren lösen zu können.

Die in der Systemkonzeption beschriebenen Geometrie- und Optimierungsmodelle des Zerstörungsreichen und des Zerstörungsarmen Modells werden in Kapitel 6 in mathematische Modelle überführt, die unabhängig von spezifischen Optimierungsverfahren angewandt werden können. Die mathematische Beschreibung ermöglicht die Entwicklung von Prototypen und die Evaluation der Modelle in unterschiedlichen Optimierungsumgebungen.

In Kapitel 7 werden inkongruente Optimierungsumgebungen und -verfahren verwendet, um die Leistungsfähigkeit und Performance der mathematischen Modelle zu testen. Die abschließende Bewertung der prototypischen Umsetzung zeigt, welches Potential und welche Probleme mit dem Einsatz der entwickelten Prototypen verbunden sind.

In Kapitel 8 erfolgt eine abschließende Bewertung des Einsatzes von Optimierungsverfahren zur Lösung von Umbau- und Umnutzungsaufgaben in der Revitalisierung. Ein Szenario für die Integration derartiger Verfahren in Entwurfsunterstützende Planungswerkzeuge beendet das Kapitel.

Die Kapitel 2 bis 4 enden jeweils mit einer Zusammenfassung, die jenen Lesern, welche die Arbeit nicht vollständig lesen möchten, zur Orientierung dient. Es wird empfohlen, Kapitel 5 in Gänze zu lesen, da dort das grundlegende Konzept beschrieben wird, welches zum Verständnis der entwickelten Prototypen erforderlich ist. Kapitel 6 ist für jene Leser entbehrlich, die nicht an der mathematischen Umsetzung der Systemkonzeption interessiert sind. Kapitel 7 erläutert die Funktionalität der Prototypen. Die exemplarischen Beispiele sind hilfreich zum Verständnis der erzielbaren Performance. Ihre Komplexität steigt chronologisch an, so dass die zuletzt beschriebenen Beispiele in der Regel die volle Leistungsfähigkeit der Prototypen demonstrieren. Alle Prototypen werden abschließend bewertet, so dass dem Leser auf Grund dieser Bewertung eine Einschätzung der verwendeten Verfahren möglich ist. Auf der beiliegenden CD befinden sich der vollständige Programmcode sowie beispielhafte graphische Darstellungen der Ergebnisse der in Kapitel 7 dargestellten Berechnungen des Prototyps OPL. Eine readme.txt beschreibt, welche Voraussetzungen notwendig sind, um den Programmcode auszuführen.

## **2 Revitalisierung - Potential, Strategien, Methodik**

### **2.1 Revitalisierungspotential**

Kapitel 2 verdeutlicht, welche Bestandsbauten heute zur Revitalisierung anstehen und welche Faktoren Einfluss auf die heutige Struktur dieses Gebäudebestandes hatten. Es wird aufgezeigt, welches Revitalisierungspotential im deutschen Gebäudebestand vorhanden ist. Ferner werden Strategien vorgestellt, die als Grundlage für die Entwicklung von Werkzeugen für die Bearbeitung von Revitalisierungsaufgaben herangezogen werden können. Die Methodik, mit der sich Architekten einer Bauaufgabe widmen, bildet den Abschluss dieses Kapitels.

#### **2.1.1 Ausgangssituation**

Warum überleben Gebäude? Mit dieser überraschenden und zugleich nachdenklich stimmenden Frage beginnt ein Beitrag Uta Hasslers [Hassler, 1999], der sich dem Thema des Umbaus des Gebäudebestandes widmet. Ebenso außergewöhnlich wie die Frage sind die Antworten, die die Autorin findet. Gebäude überleben, weil sie zu schwer zum Wegtragen sind, weil sie sich gut für neue Nutzungen eignen oder weil sie in ihrer ursprünglichen Funktion gut weiternutzbar sind. Andere wiederum überleben, weil ihre Opulenz sie vor leichtfertiger Aufgabe durch ihren Eigentümer schützt. Es ist sicher ungewöhnlich, in Bezug auf materielle Dinge von Leben zu sprechen. Hasslers Frage impliziert, dass ein Gebäude während seines Bestehens verschiedene Phasen des Gebrauchs durchläuft. Diese Phasen beinhalten herkömmlich die Errichtung, die erste Nutzungsphase, die Erneuerung, die zweite Nutzungsphase und den Abbruch des Gebäudes. Die Frage impliziert aber auch, dass das langfristige Bestehen eines Gebäudes keinesfalls

als gegeben angenommen werden darf. Vielmehr ist die Frage, ob ein Gebäude überlebt, vom sorgfältigen und nachhaltigen Umgang mit dem bestehenden Gebauten abhängig. Die Antworten beinhalten die Qualitäten, die ein Bauwerk aufweisen muss, um zu überleben. Interessanterweise sind die genannten Kriterien jene, die sich im gebauten Bestand der Jahrhundertwende widerspiegeln. Solidität der Konstruktionen ermöglicht die dauerhafte Nutzung, Flexibilität gewährleistet die Veränderung der Nutzung und Massivität hilft die Zeiten von Nichtnutzung zu überleben. Tugenden also, die eine langjährige Nutzung von Bestandsbauten erleichtern, die allerdings in den Bauten der Nachkriegszeit kaum anzutreffen sind. Gebäude, deren Alter hundert oder mehr Jahre beträgt, bestechen hingegen in vielen Fällen durch ökonomische und ökologische Qualitäten, die eine in allen Bereichen erfolgreiche Weiternutzung zulassen. Doch es sind neben sachlichen auch ästhetische Qualitäten, die den Fortbestand eines Gebäudes sichern. Viele der überlebenden Gebäude sind in ihrer Gestaltung nicht dem Geschmack des Zeitgeistes unterworfen, sondern weisen ästhetische Merkmale auf, die bis heute als unumwunden schön, ansehnlich und schützenswert empfunden werden. Ein anderes Bild hingegen ergibt sich bei der Betrachtung des Gebäudebestandes der Nachkriegsjahre. Während die Bauten der Jahrhundertwende umgebaut und weitergenutzt werden, ist dieser Bestand in vielen Fällen dem Zerfall gewidmet [Djahanschah, 1999]. Es bleibt daher an anderer Stelle zu untersuchen, welche besonderen Qualitäten der jüngere Gebäudebestand des 20. Jahrhunderts aufweist, inwieweit sich diese Merkmale auf die Dauerhaftigkeit der Bauwerke auswirken und welche Maßnahmen eingeleitet werden müssen, um deren Potenzial einer sinnvollen Weiternutzung zuzuführen.

## 2.1.2 Historische Entwicklung

Im Grunde genommen spielt die historische Entwicklung des Gebäudebestandes bezüglich dessen Potentials einer Revitalisierungsmaßnahme zugeführt zu werden, keine Rolle. Prinzipiell kommt jeder Gebäudetyp für eine Revitalisierung in Frage [Jessen und Schneider, 2000a]. Die Betrachtung der historischen Entwicklung ist jedoch von großer Bedeutung für die Entwicklung von Werkzeugen, die den Planer bei der schnellen inhaltlichen Überprüfung des Umnutzungspotentials eines Gebäudes unterstützen. Die Historie der heute noch existenten Gebäude zeigt, dass ihre Entstehung von Einflüssen wie Zeitgeist, politischen und gesellschaftlichen Entwicklungen, Architekturtheorien aber auch persönlichen Vorlieben geprägt ist. Diese Faktoren nahmen Einfluss auf die Wahl der Konstruktion, sowie die funktionale und ästhetische Ausprägung der Gebäude. Heute beeinflussen sie die Anforderungen, welche an die Entwicklung von Planungswerkzeugen gestellt werden müssen. Nur wenige Bauten des 16. oder 17. Jahrhunderts sind heute noch existent. Sie spielen für die Konzeption von Planungswerkzeugen keine Rolle, da sie in der Regel einen Sonderstatus genießen und zumeist denkmalgeschützt sind. Da diese Gebäude in der Regel relativ klein sind und ihre heutige Nutzung normalerweise äquivalent zur ursprünglichen Nutzung ist, sind sie für diese Arbeit unbedeutend. Das ausgehende 18. und beginnende 19. Jahrhundert steht im Zeichen der beginnenden industriellen Revolution. Bedingt durch Materialknappheit ist die Bauweise vieler Bauwerke durch konstruktive Innovationen gekennzeichnet. Die industrielle Weiterentwicklung des Eisengussverfahrens (Coalbrookdale Bridge, 1779) und die Erfindung des Eisenbetons (J. Monier, 1867) manifestieren sich in Bauwerken, die der Industriemoderne in der 2. Hälfte des 19. Jahrhunderts zum Durchbruch verhelfen. Dieses Zeitalter steht besonders unter dem Vorzeichen der Entwicklung neuer Technologien

und Verfahren im Ingenieurbau. Das ausgehende 19. Jahrhundert folgt dem Leitmotiv der Solidität und dem Ideal langfristiger Nutzung. Zweckmäßigkeit, Festigkeit, Dauerhaftigkeit und Schönheit in der Tradition Vitruvs bestimmen die Architektur ebenso, wie hygienische Anforderungen, Feuersicherheit und Ökonomie [Hassler, 1999]. Bauwerke werden auf lange Haltbarkeit geplant und sind mehrfach verwendbar. Herausragende Beispiele finden sich im Bereich der Industriearchitektur und der Öffentlichen Bauten, wie z.B. Gebäuden für den Verkehr (Bahnhofshallen). Die Entwicklung standardisierter Materialien und Verfahren erlaubt die Herstellung von Gebäuden, deren Beständigkeit auf große Zeiträume ausgelegt ist. Die Architektur der Gründerzeit, die in etwa auf die Zeit von 1871-1914 datiert wird, ist in vielen Städten Deutschlands allgegenwärtig. Eine große Anzahl von Wohnbauten und Stadtvierteln entstand durch die rasch wachsende Bevölkerung und den Zuzug der Landbevölkerung in städtische und industrielle Ballungszentren. Noch heute ist dieser Wohnraum von besonderer Bedeutung, da die Solidität und Massivität der Bauweise sowohl ökologische als auch ökonomische Qualitäten aufweist. Das beginnende 20. Jahrhundert eröffnet neue Perspektiven durch die Entwicklung neuartiger Materialien. Qualitativ neue Strukturen werden möglich und die Standardisierung von Werkstoffen, Produktions- und Verarbeitungsverfahren wird vorangetrieben. Bis zu Beginn des 1. Weltkrieges (1914), steht diese Entwicklung noch immer unter dem Dogma der Solidität. Diese Entwicklung erfährt im Jahre 1914 einen ersten radikalen Bruch als Antonio Sant' Elia und Filippo Tomaso Marinetti das „Manifest der futuristischen Architektur“ verkünden. Darin schreibt Sant' Elia: „Ich bekämpfe und verachte: ...die Verwendung von massiven, voluminösen, dauerhaften, veralteten und kostspieligen Baustoffen.“ [Tesching, 1996] Sant' Elia's Manifest leitet die Abkehr von der ideologisch und konstruktiv geprägten Bauweise des 19. Jahrhunderts ein. Die neuen Ideale der klassischen Moderne sind Leichtigkeit, Beweglichkeit und Dynamik sowie die Aufgabe des Anspruchs auf Dauerhaftigkeit. Nach 1918 prägen die Auflösung regionaler Systeme und die Erprobung neuer Industriebaustoffe die Architektur. Heinz und Bodo Rasch [Rasch und Rasch, 1928], Le Corbusier [Corbusier, 1924] und Mies van der Rohe sind überzeugt, dass die industrielle Fertigung und der Serienbau von Häusern unumgänglich sind. Im Jahr 1929 liefert der Maschinenbauingenieur und Kunsthistoriker Sigfried Giedion die theoretische Grundlage für die endgültige Abkehr vom Leitmotiv der Solidität und Massivität [Giedion, 1929]. Auch Walter Gropius schlägt 1925 die „fabrikmäßige Herstellung von Wohnhäusern auf Vorrat“ vor. Das „Neue Bauen“ setzt auf billige und effizient hergestellte Produkte für den kontrolliert kurzen Gebrauch. Im Jahr 1936 erscheint die Erstauflage der „Bau-Entwurfslehre“ von Ernst Neufert. Sie umfasst die „...Grundlagen, Normen, Vorschriften über Anlage, Bau, Gestaltung, Raumbedarf, Raumbeziehungen, Maße für Gebäude, Räume, Einrichtungen und Geräte mit dem Menschen als Maß und Ziel“ [Neufert, 1936] und protegirt den Einsatz standardisierter Verfahren in der Architektur und im Bauwesen. Nach dem 2. Weltkrieg avanciert sie für viele Jahrzehnte zum Standardwerk der Architektenausbildung.

Der Umbau und die Weiternutzung existenter Gebäude hat in den 30er-Jahren nur untergeordnete Bedeutung. Zur Sanierung gelangen nur wenige Gebäude, deren Lebensalter im Durchschnitt 60 Jahre beträgt. Wohl auf Grund des anhaltenden rückläufigen Neubautrends und der damit für die Architekten verbundenen Krise publiziert Konstanty Gutschow in dieser Zeit das Buch „Umbau“ [Gutschow und Zippel, 1932; nach: Hassler, 1999]. Gutschow prognostiziert darin, dass der Umbau in den kommenden Jahren an Bedeutung gewinnen wird. Und tatsächlich finden in dieser Zeit Umbaumaßnahmen statt, die in der Regel jedoch Anpassungen an den neuen

Zeitgeist und Überformungen des Alten durch das Neue sind. Die Entdekorierung des Bestandes und die Entkleidung von Ornamentik an den Bauten des 19. Jahrhunderts machen deutlich, dass eine Veränderung der Lebensumstände und damit einhergehende Bedürfnisse zur Erneuerung des Bestandes führten. Nur selten war eine Erneuerung auf Grund bautechnischer Mängel oder Bauschäden notwendig [Hassler und Kohler, 2000]. Gutschow führt ebenfalls aus, dass der Grund, weshalb Architekten sich nur selten mit Umbaumaßnahmen beschäftigten, der scheinbar damit einhergehende Verzicht „... auf eigene schöpferische Gestaltung“ sei.

Die Zeit von 1930-1945 verknüpft programmatisch Kurzzeitdesign, Rationalisierung, Ästhetik der Moderne und Monumentalität [Hassler, 1999]. Die Bauten aus der Zeit von 1919-1948 spielen heute bezogen auf ihre Gesamtnutzfläche eine untergeordnete Rolle im Gebäudebestand. Festzuhalten ist jedoch, dass wie in den nachfolgenden Jahrzehnten, der Anteil der Einfamilienhäuser und kleinen Mehrfamilienhäuser außergewöhnlich hoch ist. Dennoch üben die Bauten der NS-Zeit großen Einfluss auf das Baugeschehen nach 1945 aus. Solidität steht nun in enger Beziehung zum Ewigkeitsanspruch der Bauten des Nationalsozialismus. Das Prinzip der Dauerhaftigkeit wird, durch den Missbrauch des Prinzips in der NS-Zeit, verworfen. In den Jahren nach 1950 werden die Industrialisierungsgedanken der 20er-Jahre wieder aufgegriffen. Normierung und Effizienzsteigerung werden zu Prinzipien des Wiederaufbaus. Große Bestände der Gründerzeitjahre wurden im Krieg zerstört, so dass neuer Wohnraum für die Bevölkerung in den Städten geschaffen werden muss. Der funktionale Aufbau der Stadt wird zum planerischen Leitbild. Es entstehen Neubauquartiere in den Randbereichen der Großstädte, die getrennt nach Funktionen wie Wohnen, Arbeiten, Gewerbe und Industrie dem Prinzip der gegliederten und aufgelockerten Stadt folgen [Hassler und Kohler, 1999]. Landwirtschaftliche Flächen werden für Industrie und Gewerbe erschlossen und aus dem ehemals städtischen Kontext entfernt. Der Baubestand in den Städten wird dezimiert, um Platz für Neubauten zu schaffen. Die Ölkrise im Herbst 1973 führt erstmalig nach dem 2. Weltkrieg zum Umdenken im Umgang mit dem Gebäudebestand. Das wachsende Bewusstsein für die Endlichkeit der Ressourcen bringt neue Leitbilder hervor, welche die Reparatur der alten Stadt durch die Begrenzung von Flächenverbrauch und die Dezimierung der Suburbanisierung erreichen sollten. Unter dem Stichwort der „Nachverdichtung“ wird städtische Funktionsmischung erneut proklamiert. Durch die Verwendung von Technik und die Ratifizierung besserer Richtlinien für Neubauten glaubt man, eine adäquate Antwort auf die zunehmende Verknappung der Ressourcen zu finden. Nach 1990 und bis heute setzt sich ein Trend fort, der fast deckungsgleich mit den Entwicklungen der 20er- und 30er-Jahre des vorigen Jahrhunderts ist. Der Baubestand der 30er-Jahre, steht nun nach 60 bis 70 Nutzungsjahren zur Sanierung an. Wiederum ist festzustellen, dass Gebäude „überformt“ und Umbauten nicht unter der Prämisse ressourcenschonender Konzepte durchgeführt werden. Abermals wird der Umbau unter ästhetischen, nicht aber unter vernünftigen ökologischen Gesichtspunkten betrachtet. Konzepte wie Weiternutzung unter geringstmöglichen Eingriffen in die Gebäudesubstanz, Flächen- oder Gebäudetausch (vgl. Kap. 2.2) sind wenig bekannt und spielen, wenn überhaupt, ungeliebte Nebenrollen. Die Bauwelt formuliert es deutlich: „Den großen Wurf gibt es nicht mehr, Architekten müssen sich mit der Planung im Bestand zufrieden geben“ [Bauwelt, 1998; nach: Hassler, 1999].

### 2.1.3 Kategorisierung des Gebäudebestandes

Im Jahr 1999 stellen Nachkriegsbauten mehr als 60% des existenten Gebäudebestandes in der Bundesrepublik Deutschland. Nur etwa 10% der bestehenden Bauten sind älter als 100 Jahre. Diese Gebäude und jene aus der Zeit von 1900-1930 sind stofflich unproblematisch und höchstens durch Sanierungsmaßnahmen belastet. Bauten aus der Zeit um 1950-1960 sind baustofflich weitestgehend unbelastet, erfüllen jedoch häufig nicht die Mindestanforderungen an den Wärmeschutz. Bauten von 1960-2000 sind oft hochbelastet und im Falle eines Abbruchs mit immensen Beseitigungskosten verbunden. Sie sind sowohl problematisch bei Sanierungsmaßnahmen als auch beim Abbruch. Der Neubau verzeichnet einen Trend zur Verwendung technischer Lösungsansätze unter der Prämisse des „ökologischen Bauens“. Heutige Bauten sind in der Regel hochinstalliert. Die Kosten für Haustechnik entsprechen fast den Kosten für Baukonstruktionen, besitzen aber viel kürzere Halbwertzeiten. Handwerkliche Konstruktionen werden durch neuartige Montage- und Klebematerialien, deren Langzeitverhalten heute noch nicht absehbar ist, verdrängt. So wird „...die Lebenserwartung von Gebäuden eher kürzer, die Baustoffvielfalt größer, die Reparaturfähigkeit geringer, das Langzeitverhalten unsicherer.“ [Hassler, 1999, S. 53] Ein beträchtlicher Anteil des heutigen Gebäudebestandes (Referenzjahr 1991) stammt aus der Zeit von 1871-1918 [Kohler, 1999]. Insbesondere Handels- und Lagergebäude, Fabrik- und Werkstattgebäude, landwirtschaftliche Betriebsgebäude und sonstige Nichtwohngebäude in einer Größenordnung von ca. 600.000 Mio. m<sup>2</sup> sind aus dieser Zeit im deutschen Gebäudebestand zu verzeichnen. Auch Einfamilienhäuser und kleinere Mehrfamilienhäuser der Gründerzeit sind mit einem Anteil von ca. 290.000 Mio. m<sup>2</sup> stark vertreten. Für eine detaillierte Betrachtung siehe [Kohler, 1999].

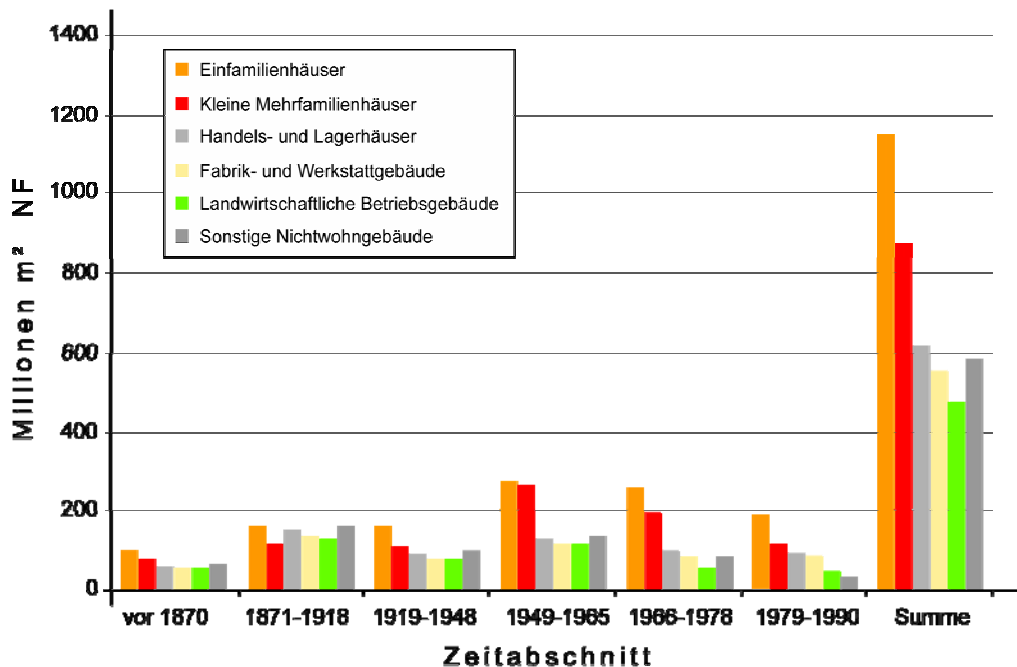


Abb. 2.1 Der deutsche Gebäudebestand im Referenzjahr 1991 in Anlehnung an [Hassler und Kohler, 2000]

Obwohl die Kategorie der Wohnhausbauten einen verhältnismäßig hohen Anteil im Bestand zu verzeichnen hat, kommt den übrigen Bauten auf Grund ihrer Komplexität besondere Bedeutung für die Entwicklung von Planungswerkzeugen zu. Die Revitalisierung von Wohnbauten (speziell Einfamilienhäuser und kleine Mehrfamilienhäuser) stellt aus planerischen Gesichtspunkten keine besondere Schwierigkeit dar, da sie in der Regel von minderer Komplexität sind und ihre Nutzungsart selten Veränderungen unterlegen ist. Der Gebäudebestand aus den Nachkriegsjahren kommt vermehrt in die erste Sanierungsphase und ist, auch auf Grund seines architektonischen Erscheinungsbildes, eine besondere Herausforderung.

Der deutsche Gebäudebestand ist bisher nur unzureichend erfasst worden. Mit Ausnahme der Wohngebäude gibt es wenig detaillierte Angaben über Gebäudetypologien. Den existierenden Erhebungen [Kohler, 1999] ist zu entnehmen, dass die Nutzungsart des Gebäudebestandes, der nicht Wohnzwecken dient, ausnehmend heterogen ist. Dennoch gibt es im Bestand Faktoren, die von der Nutzungsart unabhängig sind und Gemeinsamkeiten aufweisen. Diese Faktoren finden sich vor allem im Bereich der Konstruktionen. So zeigt die historische Entwicklung des Bestandes, dass Gebäude aus der Zeit bis 1918 in der Regel sehr solide und massiv erstellt worden sind. Umbauten dieser Gebäude sind daher an Bedingungen geknüpft, die stark durch die existente Raumstruktur geprägt sind. Bauten der Nachkriegszeit stellen einen deutlich differenzierteren Typus dar. Die Variabilität der Bauweisen und Tragkonstruktionen ist vielfältiger und ermöglicht größeren Spielraum für Umbauten. Diese Variabilität führt bei realisierten Umbaumaßnahmen häufig zu Entkleidungen des Bestandes bis auf die tragende Konstruktion und somit zum Gegenteil ökologisch sinnvoller Revitalisierungsmaßnahmen. In allen Epochen finden sich Bauwerke, die über große Nutzflächen verfügen. Auch bei den Wohnbauten ist dieser Trend, insbesondere in der Zeit von 1966-1978, zu verzeichnen.

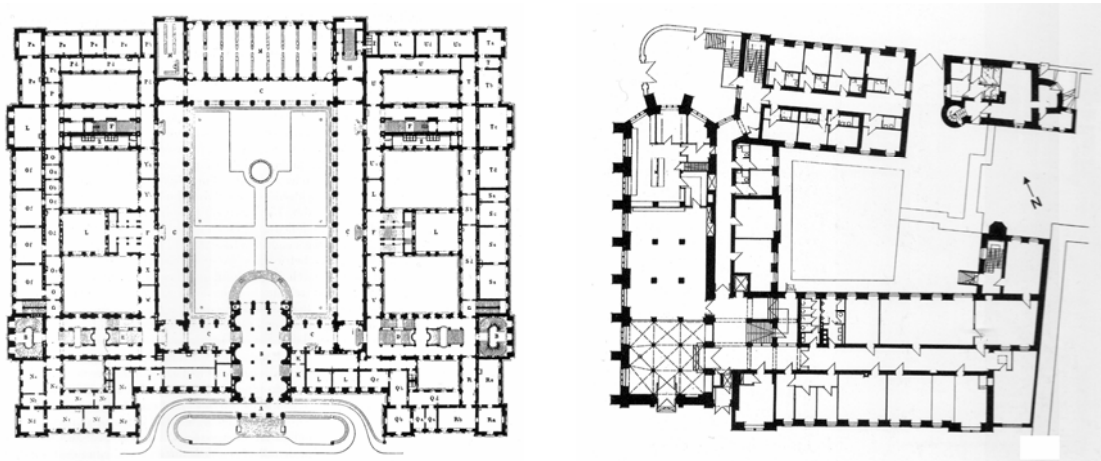


Abb. 2.2 Grundrisse eines Universitätsgebäudes und eines ehemaligen Franziskanerklosters mit massiven Tragkonstruktionen [Dolgner und Dolgner, 1996]; [Meyer-Bohe, 1991]



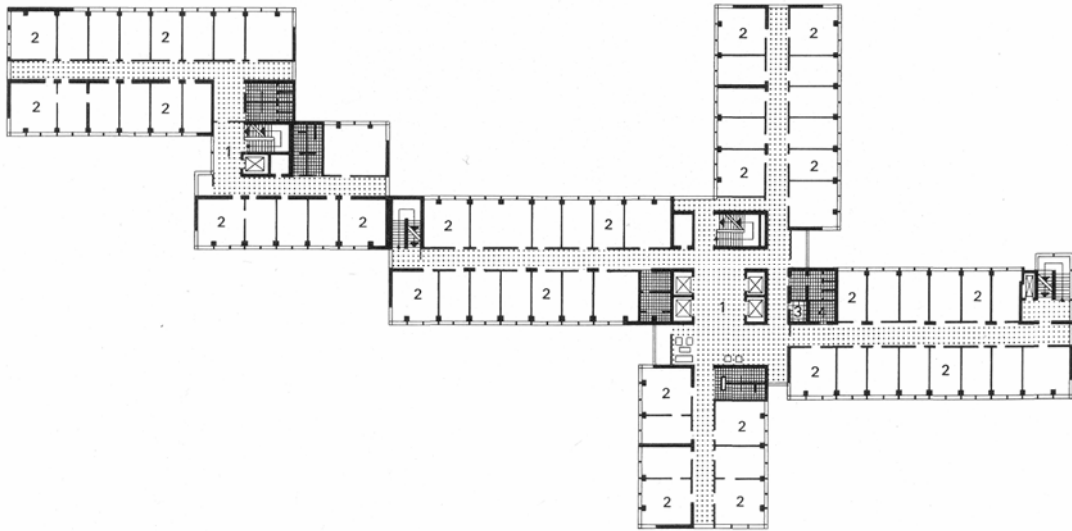


Abb. 2.3 Grundriss eines Bürohauses in Berlin (1967) in Skelettbauweise [Nagel und Linke, 1969]

### Kurzzusammenfassung

Kapitel 2.1 verdeutlicht, dass die Beschaffenheit des deutschen Gebäudebestandes und das sich für die Revitalisierung daraus ergebende Gebäudepotential eng mit der historischen Entwicklung verknüpft sind. Geprägt durch das Leitmotiv der Solidität und das Ideal der langfristigen Nutzung, entstanden gegen Ende des 19. und zu Beginn des 20. Jahrhunderts Gebäude, deren funktionale Bauweise bemerkenswerte ökonomische und ökologische Qualitäten aufweist. Bezeichnend für den überwiegenden Gebäudebestand aus den Jahren 1871-1948 sind massive Konstruktionen und daraus resultierende unflexible Grundrissorganisationen.

Die missbräuchliche Ideologisierung des Soliditäts-Dogmas während der NS-Zeit führt in den Nachkriegsjahren zur Wiederbelebung der Ideen des „Neuen Bauens“ der 20er-Jahre. Unter den Prämissen von Effizienzsteigerung und Normierung entstehen unter dem Druck des Wiederaufbaus Gebäude, deren Gebrauch auf Kurzlebigkeit ausgerichtet ist. Der Gebäudebestand der Nachkriegszeit zeichnet sich durch flexible, häufig in Systembauweise erstellte Konstruktionen und Grundrissorganisationen aus, die jedoch oft nur geringe ökonomische und ökologische Qualitäten aufweisen. Die Gebäude der Zeit von 1990-heute spielen in der Revitalisierung zum jetzigen Zeitpunkt keine Rolle. Sie stellen jedoch dahingehend einen Sonderfall dar, als dass sie sich durch die Verwendung nicht-reparaturfähiger Konstruktionen und Bauweisen von früher errichteten Gebäuden grundlegend unterscheiden.

Für die Strategien zur Revitalisierung und als Grundlage für die nachfolgende Modellbildung sind folgende Gebäude von übergeordnetem Interesse (vgl. Kap. 2.4):

- Gebäude mit unflexiblen Grundrissorganisationen und massiven Konstruktionen, jedoch guten ökonomischen und ökologischen Qualitäten (vornehmlich aus dem Zeitraum von 1871-1948)
- Gebäude mit flexiblen Grundrissorganisationen und leichten Konstruktionen, jedoch minderwertigen ökonomischen und ökologischen Qualitäten (vornehmlich aus dem Zeitraum von 1949-1990)

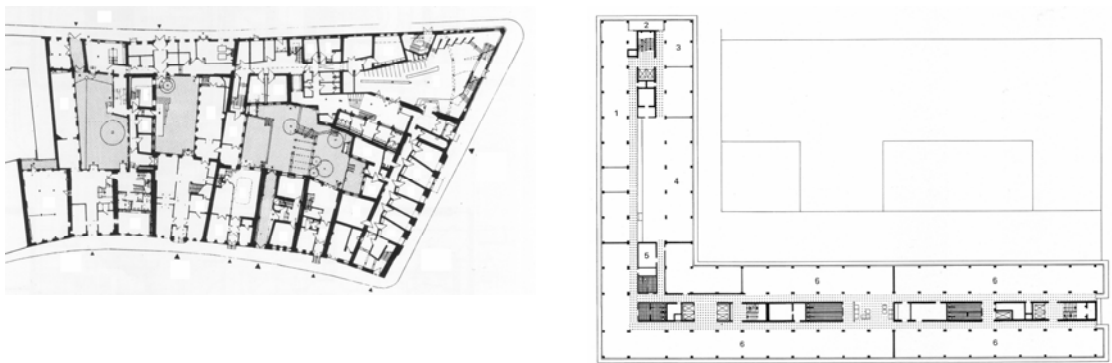


Abb. 2.4 Gebäude mit unflexibler und flexibler Grundrissorganisation [Meyer-Bohe, 1991]; [Nagel und Linke, 1969]

Zu diesen Gebäuden gehören insbesondere:

- Handels- und Lagergebäude
- Fabrik- und Werkstattgebäude
- landwirtschaftliche Betriebsgebäude
- sonstige Nichtwohngebäude

Vernachlässigbar sind Wohnbauten (Einfamilienhäuser und kleine Mehrfamilienhäuser), die zwar mengenmäßig stark vertreten sind, jedoch keine besonders komplexen Planungsaufgaben darstellen.

## 2.2 Revitalisierungsstrategien

Das übergeordnete Ziel jeder Revitalisierungsstrategie ist das ressourcenschonende Bauen und die ressourcenschonende Nutzung des Gebäudebestandes. Die zunehmende Flächeninanspruchnahme durch die Ausweisung von Bauland für Neubauten steht im Widerspruch zur Erhaltung des natürlichen Lebensraumes. Der Neubau eines massiven, vollunterkellerten Einfamilienhauses benötigt ca. 1000t Material, welches der Umwelt entnommen und verbaut wird. Beim Um-

bauen eines vorhandenen Gebäudes zu gleichen Zwecken werden im Vergleich lediglich 10t Material verbaut [Nantke, 2002]. Bei Revitalisierungsmaßnahmen entfällt die Inanspruchnahme neuer Flächen fast vollständig, da in der Regel keine zusätzlichen Flächen ausgewiesen werden müssen. Dennoch ist der Trend zum Neubau ungebrochen. So erfolgten im Jahr 1996 lediglich 5% aller Baumaßnahmen im Bestand [Losch, 1999]. Neuere Prognosen bestätigen, dass bis zum Jahr 2020 die Nachfrage nach Neubauten in Höhe von 500.000 Wohneinheiten pro Jahr gleich bleiben wird [Nantke, 2002].

## 2.2.1 Planungsgrundsätze

Die Gründe für die notwendige Auseinandersetzung mit dem bestehenden Gebauten sind vielfältig. Vor allem ökonomische, ökologische und soziale Motive bilden das Grundgerüst von Revitalisierungsmaßnahmen.

### Ökonomie

Die Triebfeder für den Aufbau und die Entwicklung städtischer Lebensräume sind und waren marktwirtschaftliche Regulationsmechanismen. Die Antwort des Architekten auf die Nachfrage des Bauwilligen nach Wohn- oder Arbeitsraum war und ist im Regelfall der Neubau. Es muss jedoch konstatiert werden, dass der Bestand an Wohn- und Arbeitsraum in der Bundesrepublik Deutschland nach der Wiedervereinigung zum größten Teil den Bedarf bereits deckt [Wang, 1999] und jede zusätzliche Schaffung von Wohn- und Arbeitsraum ein Überangebot darstellt, welches nicht nur ökonomisch sondern auch ökologisch bedenklich ist. Hinzu kommt, dass die Qualität des ungenutzten Gebäudebestandes partiell ein Niveau besitzt, welches den Vergleich zu Neubauten keinesfalls zu scheuen braucht. Insbesondere die Energiebilanz des Baubestandes der Jahrhundertwende ist durchaus in der Lage, den Vergleich mit Neubauten der zweiten Hälfte des 20. Jahrhunderts aufzunehmen [Volz, 1999]. Werden planerische Vorgehensweisen an die Besonderheiten des Bestandes angepasst, d.h. wird mit der Substanz und nicht gegen die Substanz gebaut, ist es in vielen Fällen sogar möglich, kostengünstiger einen Umbau zu realisieren als einen Neubau [Wüstenrot-Stiftung, 2000].

### Ökologie

Der Neubau ist einer der Hauptfaktoren für die Umweltbelastungen durch das Bauwesen. Dabei ist nicht allein der Primärenergiegehalt der Baustoffe und Bauprozesse relevant, sondern vielmehr der hohe Anteil an Problemstoffen, die in den Bauprozess eingebracht werden. Besonders kritisch müssen die Baubestände der Nachkriegszeit betrachtet werden. Insbesondere die Verwendung von Bauhilfsstoffen in den vergangenen Jahrzehnten führt zu einer bisher in ihrer Tragweite noch nicht absehbaren Belastung der Umwelt mit Problemstoffen. Die Bauten des beginnenden 20. Jahrhunderts spielen hier eine subordinierte Rolle, da ihre Revitalisierung - auf Grund ihrer konstruktiven und ästhetischen Beschaffenheit - vergleichsweise einfach und unproblematisch ist. Die Erhaltung und Weiternutzung des existierenden Gebauten ist unter ökologischen Gesichtspunkten sinnvoll, da die aus dem Bauprozess resultierenden und bereits eingebrachten Belastungen schon erfolgt sind und somit auf eine längere Nutzungsdauer verteilt

werden können. Ökologisch Bauen bedeutet demnach die Reduzierung des Neubauvolumens und die Verringerung des Einbringens neuer Bauprodukte.

### Soziale Gründe

Die Tendenz zur Zersiedelung und Suburbanisierung der Städte ist nach wie vor anhaltend [Friege, 1999]. Obwohl innerstädtische Brachflächen für Wohn- und Arbeitszwecke genutzt werden können, wandern sowohl die Bevölkerung als auch die Eigentümer und Nutzer von Arbeitsstätten häufig ins Umland ab [Losch, 1999]. Die Stärkung städtischer Strukturen und die Begrenzung der Abwanderung sind jedoch wesentliche Gesichtspunkte zur Förderung innerstädtischen Lebens [BMVBW, 2002] und zur Begrenzung der aus den Mobilitätsanforderungen resultierenden Umweltbelastungen. Ein Großteil der, insbesondere in den neuen Bundesländern existenten innerstädtischen Brachflächen bleibt somit ungenutzt und führt zur Verschwendung energetischer Ressourcen. Auf politischer Seite wird die Förderung des Bauens im Bestand noch nicht in ihrer Bedeutsamkeit erkannt. So gibt es nach wie vor ausreichend Förderungsinstrumente für den Neubau, wohingegen die Erneuerung des Bestandes erst in jüngster Zeit finanziell gestützt wird.

### 2.2.2 Revitalisierungsstrategien

Es gibt eine Reihe politischer Maßnahmen, die getroffen werden können, um das Planen und Bauen im Bestand zu protegieren. Die Erhebung von Abgaben auf die Inanspruchnahme natürlicher Bodenflächen und die Vergabe von Fördermitteln für den Bestand statt für den Neubau, sind wichtige Voraussetzungen, um ein Umdenken im Planungsprozess zu initiieren. Die Nutzung innerstädtischer Brachflächen hat neben ökologischen auch soziale Vorteile gegenüber der Zersiedelung der Städte durch die Ausschreibung von Bauland in den Vorstädten.

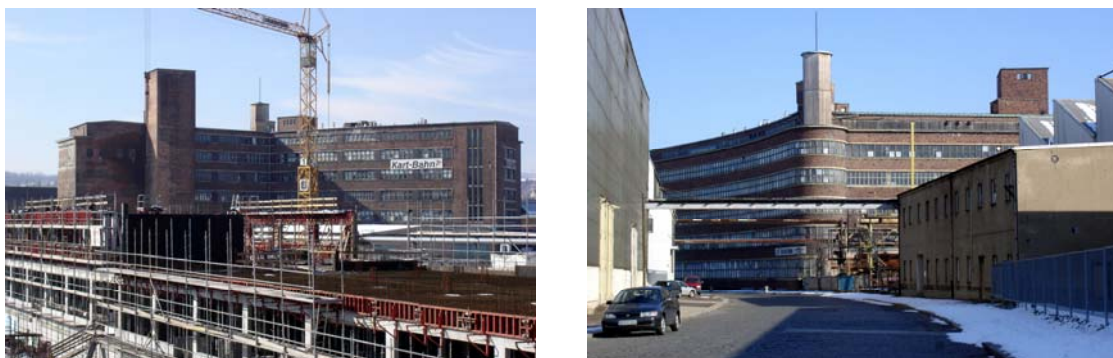


Abb. 2.5 Die gängige Vorgehensweise - Neubauen statt Umdenken

Für die planerische Tätigkeit bei der Revitalisierung von Gebäuden gibt es bislang keine allgemeingültig anwendbaren Strategien. Zwar existieren Ansätze, die exemplarisch die Vorgehensweise bei Revitalisierungsprojekten beschreiben, diese erweisen sich jedoch als auf den Einzel-

fall bezogene Vorgehensweisen, d.h. sie besitzen Gültigkeit für ein bestimmtes Projekt oder eine bestimmte Form der Gebäudenutzung und lassen sich nicht grundsätzlich verallgemeinern. Des Weiteren beinhalten diese Ansätze auch jeweils ideologische Momente, d.h. sie sind geprägt von einer bestimmten Sichtweise, die der Entscheidung für eine Revitalisierung zu Grunde gelegt wird. Diese Ideologien sind beispielsweise wirtschaftlicher, denkmalpflegerischer oder auch umweltspezifischer Natur (vgl. Kap. 2.2.2.1). Allgemein anerkannte Strategien für die Durchführung einer Revitalisierungsmaßnahme existieren nicht. Empfehlungen und Planungsgrundlagen zum Neubau und zum Bauen im Bestand finden sich u.a. bei [BMVBW, 2000], [BMVBW, 2001]. In Anlehnung an Kohler können 3 unterschiedliche Strategien als potentielle Vorgehensweisen bei Revitalisierungsmaßnahmen determiniert werden [Kohler, 1999].

#### Revitalisierungsstrategie Erläuterung

Umbau	Ein Gebäude wird durch bauliche Veränderungen an jene Anforderungen angepasst, die an einen vergleichbaren Neubau zu stellen wären. Diese Vorgehensweise ist die gebräuchlichste Art der Revitalisierung. Problematisch ist die bauliche Veränderung des Objektes hinsichtlich ökologischer Planungsgrundsätze, da in der Regel massive Eingriffe in die Gebäudesubstanz erfolgen (vgl. Kap. 2.2.2.1).
Umnutzung	Ein Gebäude wird hinsichtlich der Nutzungsart verändert. Bauliche Veränderungen werden nicht oder nur in geringfügigem Maße durchgeführt. Prinzip dieser Vorgehensweise ist der Flächen- oder Gebäudetausch. Ökologisch ist die Umnutzung sinnvoll, jedoch selten vollständig zu realisieren, da die Vor- und Nachnutzung im Idealfall deckungsgleich sein muss (vgl. Kap. 2.2.2.2).
Bauverzicht	Entmaterialisierung von Gebäuden, d.h. Funktionen die ursprünglich in Gebäuden ausgeübt worden sind, werden durch Mediennutzung kompensiert. Beispiele sind Virtuelle Unternehmen. Diese Vorgehensweise ist zwar sinnvoll hinsichtlich der Dezimierung des Neubaufkommens, reduziert jedoch nicht die Problematik der notwendigen Weiternutzung existenter Gebäudesubstanz (vgl. Kap. 2.2.2.3).

#### 2.2.2.1 Strategie 1 - Umbau

Diese Revitalisierungsstrategie beruht auf der Auflösung des existenten Gebäudes in strukturelle und/oder organisatorische Einheiten. Im planerischen Idealfall lässt sich die Gesamtstruktur eines Gebäudes in Tragwerk, Raumbildende Ausbauten und Gebäudeeinrichtung, respektive Primär-, Sekundär- und Tertiärstruktur, unterteilen. Bei Gebäuden, die in Systembauweise erstellt worden sind, ist diese Unterteilung theoretisch problemlos möglich, so dass diese Strukturen unabhängig voneinander umgeplant bzw. umgebaut werden können. Mit diesen Maßnahmen gehen in der Regel jedoch größere bauliche Eingriffe einher, die zum vollständigen Abbruch von Gebäudebauteilen bis auf die tragende Struktur führen können. Die Möglichkeit, bauliche Eingriffe in die Gesamtstruktur vornehmen zu können, führt daher oft zu Planungslösungen, die sich nicht am Bestehenden, sondern an den Visionen für das Zukünftige orientieren. In der Rea-

lität zeigt sich jedoch auch, dass bei einem Großteil der Bestandsbauten Konstruktion und Funktion eng miteinander verknüpft sind, und somit der Eintritt in eine neue Lebenszyklusphase unter veränderten Nutzungsbedingungen und Anforderungen an die Organisationsstruktur erschwert wird. Die Beschaffenheit des Gebäudebestandes der Jahrhundertwende verdeutlicht allerdings, dass gerade diese Gebäude Qualitäten aufweisen, die unmittelbar durch die nicht existente Trennung von Funktion und Konstruktion hervorgerufen werden. Dass die strukturelle Trennung für die Erneuerungsstrategie zukünftiger Bestandsbauten dennoch sinnvoll ist, zeigen aktuelle Beispiele öffentlicher Neubauvorhaben [Macchi, 1999; Kamm, 2005; Scheidegger, 2005]. Gebäude, bei welchen eine klare Unterteilung in Primär-, Sekundär- und Tertiärstruktur nicht möglich ist, können im Allgemeinen auf einer organisatorischen Ebene gegliedert werden. Die Dekomposition des Bauwerks kann durch graduelle Veränderung der Hauptnutzfläche, der horizontalen und vertikalen Erschließungsstruktur, der Lage der Nebennutzflächen und Funktionsflächen, der Fassaden- und Achsraster, bis hin zu einzelnen Bestandteilen des Tragwerkes, erfolgen. Diese Vorgehensweise ermöglicht sowohl die detaillierte Simulation der Erfüllung eines im Vorfeld festgelegten Anforderungsprofils an das Raumprogramm, als auch die wirtschaftliche Planung einer Umbaumaßnahme.

Schlote beschreibt ein Vorgehensmodell zur Revitalisierung von Bürogebäuden, welches auf der stufenweisen Dekomposition des Baukörpers beruht [Schlote, 2000]. Den Ausführungen ist vorangestellt, dass es eine Mustervorgehensweise für ein Revitalisierungsprojekt auf Grund der Verschiedenheit von Gebäudezustand, vorhandener Gebäudetechnik, Tragwerksystemen oder Achsmaßen nicht geben kann. Dennoch ist das vorgestellte Modell eine vielfach erprobte Vorgehensweise bei der Revitalisierung von Büroimmobilien und kann prinzipiell auch auf andere Gebäudetypologien übertragen werden. Die Vorgehensweise beruht auf der einleitenden Erarbeitung einer Bedarfsanalyse, respektive der Erstellung eines Funktions- und Flächenprogramms, welche sich ausschließlich am Bedarf des Auftraggebers orientiert und unabhängig von einem konkreten Gebäude erstellt wird. Es wird quasi eine Neubausimulation durchgeführt, die den Idealfall der Erfüllbarkeit des Gesamtbedarfs darstellt. Die ermittelten Kenngrößen dienen bei der abschließenden Bewertung als Benchmark-Kennziffern für Realisierungs-Alternativen durch Revitalisierungsmaßnahmen. Der einleitenden Bedarfsanalyse folgt ein grober Gebäudecheck, der sämtliche organisatorisch relevanten Detailausprägungen analysiert und Aufschluss über den Sinn einer Revitalisierungsmaßnahme gibt. In Abhängigkeit von den Ergebnissen dieser Phase folgen konkrete, stufenweise Nutzungssimulationen und deren Bewertung und Vergleich mit der Neubausimulation. Die erarbeiteten Revitalisierungsvarianten werden bezüglich der Erfüllung des Funktions- und Flächenprogramms mit dem Neubau verglichen, um unter wirtschaftlichen Gesichtspunkten die Entscheidung für eine Revitalisierung oder einen Neubau zu fällen. Durch die in drei Stufen unterteilte Klassifizierung der baulichen Eingriffe kann die Erfüllung der Anforderungen des Raumprogramms gezielt gesteuert werden. Das Vorgehensmodell für Revitalisierungsprojekte im Büroimmobiliensektor beinhaltet vier wesentliche Phasen vor der Detailplanung [Schlote, 2000].

Phase 1 - Organisatorische Bedarfsanalyse oder die Erarbeitung eines Benutzeranforderungsprofils

Tätigkeiten	Beteiligte
<p>Erarbeitung eines von einem konkreten Gebäude unabhängigen idealen Funktions- und Flächenprogramms (Flächen, Wirtschaftlichkeit, Ausstattungs-/ Nutzungsqualität etc.).</p> <p>Soweit es sich ausschließlich um die Betrachtung eines das Gebäude nicht selber nutzenden Investors handelt, sind für zu definierende Nutzerkreise Funktions- und Flächenprogramme zu simulieren.</p> <p>Bepreisung der Programmanforderungen mit Neubaukostenansätzen als notwendige Benchmark-Kennziffer</p>	<p>Nutzer/Eigentümer oder Investor mit Unterstützung eines standort erfahrenen Immobilienmaklers</p> <p>Büroorganisator</p>

Phase 2 - Durchführung eines ersten groben Gebäudechecks für das zu revitalisierende Gebäude

Tätigkeiten	Beteiligte
<p>Prüfung der Umfeldfaktoren</p> <p>Prüfung baurechtlicher und sonstiger Auflagen</p> <p>Quantitative und qualitative Grobprüfung der Flächenstruktur</p> <p>Kosteneinschätzung blockdiagrammatische Nutzungssimulation</p>	<p>Nutzer/Eigentümer oder Büroorganisator</p> <p>Optional kann es in dieser Phase bereits erforderlich sein, Fachberater einzuschalten (Brandschutzexperte, Jurist, etc.).</p>

Phase 3 - Durchführung von konkreten Nutzungssimulationen

Tätigkeiten	Beteiligte
<p>Durchführung von Nutzungssimulationen im vorhandenen Gebäude zur Bestimmung des Erfüllungsgrads der Programmanforderungen:</p> <p>Stufe 1:</p> <p>geringe bauliche Eingriffe:</p> <p>Verändert werden im Wesentlichen:</p> <p>die horizontale Erschließungsstruktur,</p> <p>die Lage/Position und Qualität von Trennwänden,</p> <p>die Qualität der Beleuchtung,</p> <p>die medientechnische Versorgung der Arbeitsplätze.</p> <p>Konstant gehalten werden:</p> <p>die vertikale Erschließungsstruktur,</p> <p>die Lage der Nebennutzflächen (u.a. Toiletten),</p> <p>der gesamte Bereich der Fassade, mit Ausnahme von inneren Sonnenschutzsystemen</p> <p>die gesamten Funktionsflächen (u.a. Technikflächen, Schächte etc.) sowie</p>	<p>Büroorganisator</p>

<p>die Brandabschnitte.</p> <p>Stufe 2: erhebliche bauliche Eingriffe (wie Stufe 1)</p> <p>Verändert werden zusätzlich: die Lage und Position der Nebennutzflächen, die Lage und Dimension der Brandabschnitte, die technischen Einbauten innerhalb einzelner Funktionsflächen, wie z.B. die Luft- und Kältetechnik (Erneuerung), Aufzüge etc.</p> <p>Weiterhin konstant verbleiben: die Position und Dimension der vertikalen Erschließung, Fassade, Achsraster.</p> <p>Stufe 3: umfangreiche Eingriffe in die Gebäudesubstanz (wie Stufe 2)</p> <p>Verändert werden zusätzlich: die vertikale Erschließungsstruktur, die Fassade.</p> <p>Weiterhin konstant verbleiben: wesentliche Teile des Tragwerks.</p>	
--	--

Phase 4 - Bewertung der Nutzungssimulationsvarianten

Tätigkeiten	Beteiligte
<p>Nutzungstechnische / organisatorische Bewertung der Varianten.</p> <p>Ausschluss von Alternativen, soweit wesentliche Funktions- und Flächenanforderungen unerfüllt bleiben, bzw. keine deutliche Situationsverbesserung erfahren.</p> <p>Bildung einer Rangreihenfolge</p> <p>Integrative fachübergreifende Grobprüfung hinsichtlich der Machbarkeit und der Kostenrelevanz:</p> <p>Gebäudesubstanz (Betonqualität, Korrosionen, Bauphysik etc.)</p> <p>Technische Gebäudeausrüstung</p> <p>Brandschutzqualität</p> <p>Architektonische Qualität</p> <p>Erarbeitung einer Entscheidungsvorlage und Entscheidung</p>	<p>Büroorganisator</p> <p>Nutzer / Eigentümer</p> <p>Statiker</p> <p>TGA-Planer</p> <p>Architekt</p> <p>Büroorganisator</p> <p>Nutzer / Eigentümer</p> <p>ggf. finanzierendes Institut</p> <p>Büroorganisator</p>



Phase 5 - Detailplanungsphase

Tätigkeiten	Beteiligte
Die Phasenausgestaltung und der beteiligte Kreis sind abhängig von dem Umfang des Eingriffs in die Gebäudesubstanz. Zu beachten ist, dass spezifisch bei Revitalisierungsprojekten die Kostenunsicherheit anfänglich höher ist als bei Neubauprojekten. Planerbeauftragungen sollten grundsätzlich in Phasen gefasst und "Ausstiegspunkte" für unkalkulierte Fälle definiert werden. Das spezifische Realisierungs-Know-how kompetenter Bauunternehmer sollte möglichst bereits in der Detailplanungsphase integriert werden. (Problem: Vertragsgestaltung!)	

Phase 6 - Realisierungsphase

Tätigkeiten	Beteiligte
Organisatorisch hat diese Phase keine gravierend anderen Abläufe gegenüber Neubauerrichtungen, daher wird auf eine Darstellung verzichtet.	

Tab. 2.1 Vorgehensmodell für Revitalisierungsprojekte im Büroimmobiliensektor [Schlote, 2000]

**2.2.2.2 Strategie 2 - Umnutzung (Flächen- und Gebäudetausch)**

Diese Revitalisierungsstrategie ist angelehnt an ein Instrument, welches ursprünglich in der Land- und Forstwirtschaft zur Durchführung von Flurbereinigungen eingesetzt wurde [Bundesministerium für Verbraucherschutz, 1953]. In der Architektur findet es auf Grund des bisher nicht hinreichend erkannten Potentials kaum Anwendung. Prinzipiell werden bei diesem Modell Flächen zwischen mehreren Parteien getauscht, wenn die Anforderungen des jeweiligen Eigentümers an seine eigene zu tauschende Fläche nicht erfüllt werden können und der Erwerb der getauschten Fläche die Erfüllung dieser Ansprüche ermöglicht. Es ist offensichtlich, dass die Anforderungen der einzelnen Parteien an die jeweilige Fläche unterschiedlich sein müssen, um einen sinnvollen Tausch durchzuführen. In der Architektur bedeutet Flächentausch demnach, durch Änderung des Nutzer- oder Nutzungsprofils die Schwachstellen eines bestehenden Gebäudes auszugleichen [Kahlert, 1999]. Derartige Schwachstellen sind beispielsweise die geringe Flexibilität der in Massivbauweise erstellten Verwaltungsbauten des beginnenden 20. Jahrhunderts oder die schlechten klimatischen Verhältnisse der Verwaltungsgebäude aus den 60er und 70er-Jahren, die ohne bauliche Eingriffe nur dann ausgeglichen werden können, wenn diese Schwachstellen für das neue Nutzerprofil geringe oder keine Relevanz besitzen. Der Vorteil des Flächentauschs besteht vor allem in der von der Gebäudekonstruktion unabhängigen Durchführbarkeit. Während beim Umbau (vgl. Kap. 2.2.2.1) die Auflösung eines bestehenden Gebäudes in strukturelle Einheiten mit der Frage verbunden ist, ob Primär-, Sekundär- und Tertiärstruktur voneinander trennbar sind, ist beim Flächentausch lediglich die organisatorische Ausprägung der zu tauschenden Nutzflächen und ihre Anbindung an Verkehrsflächen und Nebennutzflächen relevant. Der Flächentausch basiert auf der organisatorischen Dekomposition eines Gebäudes in Einzelflächen, die unabhängig voneinander genutzt werden können und gleichzeitig Zugriff auf - auch potentiell gemeinschaftlich zu nutzende - Nebennutzflächen gewährleisten. Der Flächentausch eignet sich daher für alle Gebäudetypen mit ausreichend großer Nutzfläche und entsprechenden organisatorischen Unterteilungsmerkmalen. Getauscht werden sowohl organisatorisch zusammenhängende Raumeinheiten und Geschosse, als auch einzelne Räume. Der Flächentausch kann ebenfalls auf Raumeinheiten, die sich in unterschiedlichen

Geschossen befinden, angewendet werden. Das vorgenannte gilt gleichermaßen für den Gebäudetausch, bei welchem ganze Gebäude oder Gebäudegruppen getauscht werden können. Beide Tauschmodelle eignen sich insbesondere für Institutionen oder Firmen mit großem Gebäudebestand. Neben den positiven ökologischen Auswirkungen auf den Gebäudebestand, ist der Flächen- und Gebäudetausch daher auch ein wirtschaftliches Instrument zur Verwaltung und Nutzungsoptimierung von Liegenschaften im Facility Management.

### **2.2.2.3 Strategie 3 - Bauverzicht**

Auch der Bauverzicht stellt heute in Teilbereichen eine mögliche Lösung zur Verringerung des Neubauvolumens dar. Wenn Unternehmensprozesse von Gebäuden losgelöst werden können, ist es möglich, auf die Errichtung dieser Gebäude ganz oder teilweise zu verzichten. Insbesondere die Nutzung von Informations- und Kommunikationssystemen ermöglicht eine derartige Verlagerung von Prozessen [Anders, 1999; Donath, Lömker et al., 2001]. Im Wesentlichen beziehen sich diese Prozesse auf Geschäftsvorgänge. Durch die Bildung Virtueller Unternehmen, können rechtlich eigenständige Firmen Zusammenschlüsse bilden, die nach außen hin eine scheinbar geschlossene Einheit bilden, deren Einzelunternehmen jedoch regional oder global verteilt sind. Die Kommunikation zwischen den Unternehmen basiert auf einer leistungsstarken technischen Infrastruktur, insbesondere auf der Nutzung digitaler Medien zur Projektorganisation und Projektdurchführung. Derartige Ansätze sind seit einigen Jahren in öffentlichen Einrichtungen (z.B. Bibliotheken und Verwaltungen) zu finden, in denen die physische Präsenz des Nutzers oder des Dienstleistungsanbieters nicht notwendigerweise erforderlich ist.

Es ist jedoch festzuhalten, dass der größte Teil der Virtuellen Unternehmen in ihrer jetzigen Struktur keinen nennenswerten Beitrag zur Verringerung des Neubauaufkommens leistet. Es muss auch konstatiert werden, dass die Existenz und der Ausbau dieser Unternehmensform sogar negative Auswirkungen auf die Weiternutzung des Gebäudebestandes haben kann. Die Gründung neuer Unternehmen mit geringem Startkapital war in der Vergangenheit mit der Notwendigkeit verbunden, kostengünstige Räumlichkeiten anzumieten. Diese wurden in der Regel im unsanierten Gebäudebestand gefunden. Setzt sich der Trend zur Nutzung von IuK-Technologien fort, entfällt möglicherweise dieser gewichtige Faktor der Weiternutzung des Gebäudebestandes. Es existieren allerdings auch Organisationsformen von Unternehmen, deren Geschäftsabläufe fast vollständig durch Mediennutzung abgewickelt werden können. Auktionshäuser, wie die Firma ebay, die vor wenigen Jahren Neubauten für Lagerhaltung hätten errichten müssen, können heute durch neuartige Vertriebswege auf deren Existenz vollständig verzichten. Die Methodik des Bauverzichts durch Entmaterialisierung ist nicht eindeutig positiv zu werten, da zwar einerseits die Verringerung des Neubauvolumens ermöglicht wird, andererseits jedoch nicht zwingend die Weiternutzung von Bestandsbauten erforderlich ist. Die Bauverzichtstrategie kann also nur dann als sinnvolle Revitalisierungsstrategie in Betracht gezogen werden, wenn Unternehmensprozesse, die den Neubau von Gebäuden zwingend erfordert hätten, durch Mediennutzung kompensiert werden.

Die Bauverzichtstrategie ist jedoch nicht nur aus dem Blickwinkel des vollständigen Verzichts auf das Neubauen oder Umbauen zu betrachten. Die organisatorische Neuordnung von Bauwerken durch Umbau oder Umnutzung wird selten mit der vollständigen Erfüllung eines an einer Neubausimulation orientierten Raumprogramms einhergehen. Folgerichtig besteht unter dieser

Prämisse die Gefahr, dass hier der Neubau präferiert wird. Durch die Teilkompensierung von Prozessen mit Hilfe digitaler Medien, oder durch teilweisen Bauverzicht auf bestimmte Flächeneinheiten wie Nebennutzflächen oder Funktionsflächen für jede in sich abgeschlossen nutzbare Raumeinheit, können Defizite dieser Modelle kompensiert werden. Diese Vorgehensweise erfordert ein Umdenken und möglicherweise auch einen verminderten Anspruch an die Nutzungsqualität eines bestehenden Gebäudes. Dass darin ökologisches und ökonomisches Potential enthalten ist, zeigen programmatisch verwandte Ansätze bei Neubaulösungen wie bspw. die Entwicklung der Kombi-Büros [Congena, 1994].

Der Bauverzicht wird nachfolgend nicht weiter vertieft, da diese Strategie eine von dieser Arbeit differente Forschungsthematik darstellt.

### **Kurzzusammenfassung**

Die Entscheidung, ob und wie ein Gebäude revitalisiert wird, wird in der Regel unter ökonomischen, ökologischen und sozialen Gesichtspunkten getroffen. Fällt diese Entscheidung gegen eine Revitalisierung aus, verbleiben als Alternative der Neubau oder der Verzicht auf das Bauen. Der Bauverzicht kann dann erfolgen, wenn die Funktion, die ursprünglich einem Gebäude zugeordnet worden ist, durch Mediennutzung kompensiert werden kann. Fällt die Entscheidung für eine Revitalisierung aus, kann ein Gebäude umgebaut oder umgenutzt werden. Der wesentliche Unterschied besteht darin, dass der Umbau immer die bauliche Veränderung einschließt. Die Umnutzung kann theoretisch ohne bauliche Veränderungen erfolgen.

Der Umbau basiert auf der strukturellen und/oder organisatorischen Dekomposition eines Gebäudes in zusammengehörige Einheiten. Diese Einheiten können unabhängig voneinander umgeplant und umgebaut werden. Umbau-Maßnahmen bedingen in der Regel größere bauliche Eingriffe. Die Umnutzung basiert auf dem Tausch von Flächen innerhalb des Gebäudes. Diese Maßnahmen bedingen keine oder nur geringe bauliche Eingriffe. Beide Strategien besitzen Gemeinsamkeiten. Die organisatorische Analyse des Gebäudes und des zu integrierenden Raumprogramms ist in beiden Fällen notwendig. Die Anwendung der Strategien wird in vielen Fällen kombiniert erfolgen und kann für nahezu jeden Gebäudetypus eingesetzt werden. Das Prinzip des Flächentauschs ist gleichermaßen auf den Gebäudetausch anwendbar. Folgende Faktoren sind festzuhalten:

- Der Umbau ist besonders für die Bauten der Nachkriegszeit mit ihren flexiblen Grundrissorganisationen und leichten Konstruktionen geeignet. Diese Bauten können mit geringeren Eingriffen in die Bausubstanz in größerem Umfang verändert werden, da ihre Primär-, Sekundär- und Tertiärstruktur in der Regel trennbar ist.
- Die Umnutzung durch Flächentausch ist hauptsächlich für die Bauten der Vorkriegszeit, die unflexible Grundrissorganisationen und massive Konstruktionen aufweisen, geeignet. Diese Bauten können nur mit erheblichen Eingriffen in die Bausubstanz in größerem Umfang verändert werden, da ihre Primär-, Sekundär- und Tertiärstruktur in der Regel nicht trennbar ist.
- Die Umnutzung ist dem Umbau bei allen Gebäudetypen vorzuziehen, da diese Strategie theoretisch keine oder nur geringe Eingriffe in die Bausubstanz nach sich zieht.

## 2.3 Revitalisierungsmethodik

Bei der Bearbeitung einer Revitalisierungsaufgabe ist eine methodische Vorgehensweise ebenso unabdingbar, wie bei anderen Planungsaufgaben in der Architektur. Dieser Umstand wirft die Frage auf, wie Architekten planen und entwerfen und welche Methodik sie benutzen, um eine planerische Aufgabe zu lösen. Es soll hier nicht der Versuch unternommen werden, eine allgemeingültige Planungsmethodik für die Revitalisierung aufzustellen. Dieses Vorgehen ist, auch auf Grund des nicht eindeutig typologisierbaren, heterogenen Gebäudebestandes, unmöglich. Der Auseinandersetzung mit Planungsmethoden können jedoch Hinweise darauf entnommen werden, welche Aspekte des Planens und Entwerfens für die Bildung von Revitalisierungsmodellen Relevanz besitzen. Da keine einheitliche Begriffsbestimmung der Begriffe „Planung“ und „Entwerfen“ existiert, werden die Begrifflichkeiten und ihre Verwendung im Kontext dieser Untersuchung geklärt.

### 2.3.1 Planen und Entwerfen

„Jedermann entwirft manchmal; niemand entwirft immer. Entwerfen ist nicht das Monopol derjenigen, die sich selbst „Planer und Entwerfer“ nennen. Vom Stadtentwicklungsplan bis zum elektronischen Schaltkreis, vom Steuergesetz bis zur Marketingstrategie, von der „strategischen Verteidigungsinitiative“ bis hin zur Einkaufsliste für das Mittagessen am kommenden Sonntag, es sind alles Ergebnisse von Plänen, die aus der „Entwerfen“ genannten Tätigkeit resultieren.“ [Rittel und Reuter, 1992, S. 135]

Bereits in den 30er-Jahren wurde das Thema „Planung“ in ersten Veröffentlichungen diskutiert. Die Auseinandersetzung mit der im englischsprachigen Raum „design methods“ genannten Thematik, begann in den frühen 60er-Jahren. Ziel dieser Auseinandersetzung war der Wunsch, das Planen und Entwerfen besser kontrollieren zu können. Es sollte ein Instrumentarium geschaffen werden, mit dessen Hilfe die Grenzen der „Erfahrung“ des Planers und Entwerfers überwunden werden konnten, so dass der Planungsprozess nachprüfbar und die daraus abgeleiteten Ergebnisse der Planung bewiesenermaßen richtig waren. In den 90er-Jahren kam die Forschung im Bereich der Planungsmethodik im deutschsprachigen Raum fast völlig zum Erliegen. Über die Ursachen dafür kann nur spekuliert werden. Wesentlich wichtiger ist die Frage, ob die entwickelten Planungsmethoden in der Praxis überhaupt zur Anwendung gelangten, oder ob nicht jeder Architekt seine ganz eigene Herangehensweise an eine komplexe Aufgabenstellung im Bereich des „design“ hat. Schill-Fendl liefert eine umfassende Auflistung, der für den Bereich des Planens und Entwerfens existenten Planungsmethoden. Im Planungsmethodischen Glossar werden 130 Verfahren und Techniken, Instrumente und Werkzeuge als Teilmethoden dargelegt, „...mit denen der Architekt seinen Planungs- und Entwurfsprozess in der Praxis effizient gestalten kann“ [Schill-Fendl, 2004]. So werden beispielsweise Befragung, Beobachtung und Brainstorming-Techniken beschrieben, mit denen die meisten Architekten vertraut sind. Andererseits werden auch weniger bekannte Instrumente, wie Expertensysteme, Graphentheorie und Künstliche Intelligenz benannt und als „Teilmethoden zur Bewältigung komplexer Bauaufgaben“ vorgestellt [Schill-Fendl, 2004, S. 327]. Eine Umfrage unter Architekten, welche der

dargelegten Methoden tatsächlich und vor allem, ob diese im Sinne ihrer wissenschaftlichen Definition korrekt eingesetzt werden, gibt es bisher leider nicht. Diesem Defizit widmet sich Lorenz auf eine ganz andere Weise [Lorenz, 2004]. Er befragt 25 namhafte Architekten nach ihren ersten Arbeitsschritten zu Beginn einer Entwurfsaufgabe. Bezeichnenderweise nennt kein einziger Architekt explizit auch nur eine der von Schill-Fendel zusammengetragenen Planungsmethoden. Dessen ungeachtet wird aus den Antworten der Architekten deutlich, dass Entwürfe nicht aus einem kreativen Chaos erwachsen, sondern durch „...sehr klar strukturierte und durchdachte Arbeitsabläufe...“ entstehen. In Anlehnung an Günter Behnisch benennt Lorenz vier Arbeitsabläufe, die für die Mehrzahl der befragten Büros Gültigkeit besitzen: Informationsphase, kreative Phase, Bewertungsphase und Ausarbeitungsphase. Fritz Auer ergänzt, dass auch die „Recherche vergleichender Beispiele“ eine Rolle spielt, „...um sich der Bautypologie anzunähern und diese vielleicht im eigenen Entwurf weiterzuentwickeln“. Anders als die von Schill-Fendel mit Einschränkungen als Planungs- und Entwurfsmethode charakterisierte HOAI, welche dem Entwerfen nur die Leistungsphase 3 und in der gängigen Interpretation der Vorplanung als „Vorentwurfsplanung“ auch die Leistungsphase 2, einräumt, sehen die befragten Architekten mehrheitlich den Prozess des Entwerfens erst mit Fertigstellung des Gebäudes als beendet an. Auch Schill-Fendel definiert das Entwerfen als Prozess, der in allen Phasen der Planung stattfindet.

„Das Entwerfen ist ein schöpferischer (kreativer) Prozess innerhalb der und parallel zur Planung, während dessen systematisch oder intuitiv eine im Vorhinein nicht berechenbare Vorlage für ein einmaliges architektonisches Objekt zu einem bestimmten Nutzen und zur späteren endgültigen Herstellung entwickelt wird. .... Ein Entwurf ist ein vorläufiges, weiterhin fortzuschreibendes Zwischenergebnis der Planung und des Entwerfens. Aufgrund des Zusammenhangs mit der fortzuführenden Informationsbeschaffung und -verarbeitung unterliegt ein Entwurf ständigen Vor- und Rückkopplungen, Überprüfungen und Aktualisierungen.“ [Schill-Fendel, 2004, S. 45]

„Mit Planung ist ein systematischer, informationsverarbeitender Prozess zur zielführenden Lösung einer Aufgabenstellung gemeint... “. [Schill-Fendel, 2004, S. 59]

Zusammenfassend lässt sich, in Anlehnung an Schill-Fendel, eine für diese Arbeit gültige Betrachtungsweise festhalten, die zum Ausdruck bringt, dass sich Planung auf den analytisch-theoretischen Teil eines Prozesses bezieht und das Entwerfen den praktisch-synthetischen Bereich des Prozesses beinhaltet. Zur Vertiefung dieses Themengebietes sei auf die ausführliche Arbeit von Schill-Fendel verwiesen, die umfangreiche Quellen zitiert und u.a. die Arbeiten von Joedicke, Rittel und Schirmbeck, benennt.

Es ist nicht verwunderlich, dass die in wissenschaftlichen Arbeiten entwickelten Methoden und Konzepte in der Praxis eine untergeordnete Rolle spielen. Ebenso wenig überrascht es, dass kaum ein praktizierender Architekt diese Methoden in Gänze überschaut, geschweige denn gezielt anzuwenden wüsste, oder sich deren Anwendung bewusst wäre. Allein die Tatsache, dass es 130 Methoden zur Planung gibt, wird bei einer Vielzahl der Architekten auf Unverständnis stoßen. Es sei jedoch auch festgehalten, dass zumindest die Aussagen der von Lorenz befragten

Architekten die Interpretation zulassen, dass sie sich jeder fremden Planungsmethodik entziehen und ausschließlich ihre eigene, mitunter jedoch äquivalente, Planungsmethodik anwenden. Überraschend sind in diesem Zusammenhang allerdings Anmerkungen, die die gängige Praxis vollends konterkarieren. So wird die Hinzuziehung von Sonderfachleuten beispielsweise auf die von Behnisch benannte Ausarbeitungsphase verlagert. Da diese Aussage nach [Lorenz, 2004] beispielhaft für alle befragten Büros ist, wären von diesen demnach keine Entwürfe zu erwarten, die eine ganzheitliche Betrachtungsweise von Architektur und den mit ihr verbundenen Disziplinen erlauben. Da die gebauten Objekte der befragten Architekten jedoch eine andere Sprache sprechen, liegt die Vermutung nah, dass auch die eigene Planungsmethodik mitunter verlassen wird und ein Konglomerat unterschiedlicher Techniken zum Einsatz gelangt.

Die besagte HOAI gliedert zumindest den Planungsprozess und die damit verbundenen Bearbeitungsphasen dergestalt, dass klar erkenntlich ist, welchen Handlungsspielraum der Architekt im Planungsprozess besitzt. In den frühen Phasen ist dieser typischerweise charakterisiert durch ein Defizit an planungsrelevanten Informationen, gefolgt von einem sehr hoch verdichteten Informationsaufkommen gegen Ende der Vorplanung mit dem Eintritt in die Entwurfsplanung. Der Abschluss der zweiten Leistungsphase ist normalerweise entscheidend für die Weiterentwicklung eines Projektes. Für alle am Planungsprozess eines Bauwerkes in dieser Phase beteiligten Parteien ist die Entscheidung über die Fortsetzung der Planungstätigkeit auf größtmögliche Informationsvielfalt zu stützen. Für den Bauherren stehen gerade zu diesem Zeitpunkt oft wirtschaftliche Überlegungen im Vordergrund des Interesses. Hinsichtlich der Anforderungen ganzheitlicher Gebäudeplanung ermöglicht jedoch nur die frühe Planungsoptimierung ein effizientes Ergebnis. Im Sinne einer nachhaltigen Bauweise müssen die wesentlichen, für die Formfindung und Ausprägung des Bauvorhabens wichtigen Planungsschritte zu diesem Zeitpunkt bereits vollzogen sein [Rogers und Gumuchdjan, 1997]. Die Größe des Informationsangebotes gegen Ende der Vorplanung ist also für die Weiterentwicklung eines Bauvorhabens von entscheidender Bedeutung. Die erste Phase der Planungstätigkeit (Leistungsphase 1 – Grundlagenermittlung) ist bereits so komplex, dass die dem Architekten obliegende Aufklärungs- und Beratungspflicht nur in den seltensten Fällen vollständig und rechtssicher erbracht werden kann. Die Fülle der vom Architekten bereits während der Grundlagenermittlung zu berücksichtigenden Faktoren ist als Grundleistung in der Honorarordnung festgehalten. Demnach ist der Architekt verpflichtet, den Bauherren zum gesamten Leistungsbedarf zu beraten. Diese Beratung umfasst zum einen immer die Darlegung der zur Durchführung der Bauabsicht im Einzelnen nach Art und voraussichtlichem Umfang erforderlichen Unternehmerleistungen, zum anderen aber auch die Bekanntgabe des notwendigen Einsatzes von Sonderfachleuten. Der Umkehrschluss legt nahe, dass eine Beratung zum voraussichtlichen Umfang von Unternehmerleistungen nur erfolgen kann, wenn der Architekt bereits zu diesem Zeitpunkt über entsprechende Informationen zu den Rahmenbedingungen des Bauvorhabens und über konkrete Vorstellungen zur Umsetzung seiner Entwurfsidee verfügt. Das anfängliche Informationsdefizit müsste demzufolge bereits zu diesem Zeitpunkt in Gänze durch den Architekten aufgearbeitet sein. Außerdem ist bereits in dieser ersten Phase auf Richtlinien und Erlasse des Gesetzgebers hinzuweisen, welche in die Planungstätigkeit einzubeziehen sind. Für die Grundlagenermittlung ist es die im Vordergrund stehende Verpflichtung des Architekten, den Auftraggeber richtig und im jeweils gebotenen Umfang umfassend zu beraten. In der zweiten Phase der Planungstätigkeit (Leistungsphase 2 – Vorplanung) muss der Architekt die wesentlichen städtebaulichen, gestalterischen, funktionalen, technischen, bauphysikalischen,

wirtschaftlichen, energiewirtschaftlichen und landschaftsökologischen Zusammenhänge, sowie die Belastung und Empfindlichkeit der betroffenen Ökosysteme klären und erläutern. Ferner ist er verpflichtet, als Grundleistung der Vorplanung die Integration der Leistungen aller an der Vorplanung fachlich Beteiligten zu gewährleisten. Zusammenfassend handelt es sich um eine Fülle notwendiger Überlegungen mit dem Ziel, die für die Verwirklichung des Vorhabens wichtigen Anforderungen zu klären und dem Auftraggeber die gefundenen Ergebnisse zu vermitteln. Die Masse, der zu diesem Zeitpunkt als notwendig und unabdingbar anzusehenden Informationen, zwingt den Architekten ein Maß an Leistungen, Verantwortlichkeit und Aufwand zu erbringen, das der allgemeinen Bewertung und Gewichtung der gesamten Vorplanungsphase kaum angemessen erscheint. Infolgedessen werden zwangweise Lösungen präferiert, deren Funktionalität den allgemeinen Regeln der Technik entspricht, die jedoch das mögliche Potential einer alternativen Lösung nach gleichen oder grundsätzlich verschiedenen Anforderungen außer Acht lassen. Den Leistungsphasen 1 und 2 kommt in der Revitalisierung besondere Bedeutung zu. Begreift man den Revitalisierungsprozess eines Bauwerkes als lebenszyklusorientierten Prozess, an dessen Ende unter Umständen auch der ‚Nichtbau‘ als Ergebnis stehen kann, wird deutlich, wie wichtig die Bearbeitung dieser Leistungsphasen für den Architekten in der Zukunft sein wird. Entgegen der weit verbreiteten Planungspraxis, bei der während des Baufortschrittes auf der Baustelle grundlegende Planungsentscheidungen revidiert oder neu interpretiert werden, kann der nachhaltige und wirtschaftliche Umgang mit dem Bestand nur in den frühen Planungsphasen gewährleistet werden.



Abb. 2.6 Bauaufgaben der Zukunft

Da sich die Nutzung und Umnutzung des Bestandes heute aus vollkommen anderen Blickwinkeln darstellt, muss sich auch die Planungsstrategie entsprechend verändern und den gegebenen Verhältnissen angleichen. Betrachtet man Revitalisierungsmaßnahmen unter der Prämisse des Lebenszyklus eines Gebäudes wird deutlich, dass sich der Architekt mit einem ehemals funktionierenden und bereits fertig gestellten Produkt auseinandersetzen muss. Dieses existente Produkt, welches den aktuellen Erfordernissen des potentiellen Nutzers nicht mehr gerecht wird, muss den Bedingungen nachhaltiger Planung folgend, eher optimiert und reorganisiert, als lediglich saniert werden. Dabei müssen veränderte Nutzungsanforderungen mit existenten Rahmenbedingungen und Eigenheiten des spezifischen Bauwerkes abgestimmt werden. Die Aufgaben, denen sich der Architekt in Zukunft gegenübergestellt sieht, sind demzufolge Optimierungs-, Umstrukturierungs- und Erhaltungsaufgaben.

### **Kurzzusammenfassung**

Der Auseinandersetzung mit Planungsmethoden können folgende Punkte, die Einfluss auf die Bildung von Modellen für die Revitalisierung ausüben, entnommen werden:

- Planen und Entwerfen sind sich gegenseitig bedingende Prozesse, die klar voneinander abgegrenzt werden können. Das Entwerfen ist ein ergebnisorientierter, kontinuierlich zu aktualisierender Prozess, der innerhalb des Planungsprozesses stattfindet.
- Keine Planungsmethodik kann den Anspruch auf Allgemeingültigkeit erheben. Architekten verwenden unterschiedliche Methoden und Techniken zur Lösung komplexer Probleme. Diese Methoden und Techniken werden auch miteinander vermengt.
- Am Planungsprozess sind unterschiedliche Parteien, deren Leistungen vom Architekten koordiniert werden, beteiligt.
- Erinnerung, Erfahrung und Vergleich spielen wichtige Rollen bei der Entwicklung von Entwürfen.
- Der Planungsprozess erfordert die Entwicklung von alternativen Lösungsmöglichkeiten einer Entwurfsaufgabe nach gleichen oder grundsätzlich verschiedenen Anforderungen.
- Entwurfsergebnisse müssen bereits zu Beginn des Planungsprozesses schnell erarbeitet werden. Diese Erarbeitung geschieht teilweise in Unkenntnis aller relevanten Rahmenbedingungen.
- Die Entscheidung, ob eine Revitalisierungsmaßnahme durchgeführt wird, ist in entscheidendem Umfang abhängig von den Ergebnissen dieses frühen Entwurfsprozesses.
- Das Entwerfen ist nicht frei von normativen Einflüssen. Gesetze, Verordnungen und Richtlinien müssen ebenso beachtet werden, wie gestalterische, wirtschaftliche oder technische Anliegen des Auftraggebers.



## 2.4 Zusammenfassung

Am Vorgehensmodell für Revitalisierungsprojekte im Büroimmobiliensektor (vgl. Kap. 2.2.2.1) ist deutlich erkennbar, unter welchen Gesichtspunkten eine Revitalisierungsmaßnahme im Allgemeinen beurteilt wird. Primäres Ziel eines Immobilieneigners ist, die Vermietbarkeit der Immobilie sicher zu stellen und höchste Qualität in Bezug auf die zu vermietenden Räume zu gewährleisten. Aus diesem Grund ist es verständlich, dass Schlote in seinem Modell die Bewertung der zu revitalisierenden Immobilie an Hand eines Vergleichs mit einer Neubaumaßnahme durchführt [Schlote, 2000]. Diese Vorgehensweise wird in der Regel zu einem für den Investor zufrieden stellenden Ergebnis führen, da hohe Maßstäbe angelegt werden, die durch die Unterteilung in drei Stufen auch ausreichend große Spielräume für alternative Lösungsmöglichkeiten bieten. Unter dem Gesichtspunkt ökologisch sinnvoller Revitalisierungsmaßnahmen ist diese Vorgehensweise des direkten Vergleichs mit einer Neubaumaßnahme jedoch fragwürdig. Der Sinn einer derartigen Revitalisierungsmaßnahme ist die möglichst zerstörungsarme Nachnutzung, d.h. die Weiternutzung oder Umnutzung eines Gebäudes. Diese Nachnutzung unterliegt durchaus der Perspektive, dass durch den Verzicht auf bauliche Eingriffe die vollständige Erfüllung des idealen Raum- und Flächenprogramms nicht gegeben sein kann. In den wenigsten Fällen werden Revitalisierungsmaßnahmen dem Anspruch eines Vergleichs mit einer Neubaumaßnahme standhalten. Die Gefahr ist groß, dass ein derartiger Vergleich zu Gunsten eines Neubaus entschieden wird. Die Umbaustrategie verdeutlicht aber auch die Variabilität, die diese Vorgehensweise beinhaltet. Es ist möglich, die Tiefe des Eingriffs in die bestehende Struktur und damit den Grad der baulichen Eingriffe gezielt zu lenken.

Im unter ökologischen Gesichtspunkten betrachteten Idealfall des Verzichts auf bauliche Eingriffe ist es ausreichend, ein Gebäude organisatorisch zu zerlegen und durch Flächen- oder Gebäudetausch auf neue Nutzerprofile abgestimmte veränderte Nutzungseinheiten zu bilden (Umnutzungsstrategie). Die besprochenen Strategien Umbau und Umnutzung erfordern vom Architekten eine veränderte Herangehensweise an den Planungsprozess. In Folge dieser Herangehensweise ist unter Umständen keine Bauaufgabe, sondern lediglich eine kombinatorische Umstrukturierung und Neuordnung von Flächen zu lösen. Diese entspricht eher dem Aufgabenbereich des Operations Research, als dem des traditionellen Architekten. Es ist offensichtlich, dass eine derartige Aufgabe methodische Kenntnisse erfordert, die der Architekt in der Regel nicht besitzt und für deren Bearbeitung es keine digitalen Hilfsmittel in der Architektur gibt.

Noch drastischer sind die Auswirkungen des Bauverzichts auf den Planungsprozess. Dieser enthebt den Architekten vollständig seines gewohnten Instrumentariums und erfordert eine Neuorientierung im Bereich der Mediennutzung. Sie bietet dadurch jedoch ebenfalls die Chance, neue Tätigkeitsfelder zu erschließen und den Beruf des Architekten unter veränderten Bedingungen neu zu definieren.

Zusammenfassend lassen sich aus Kapitel 2.1 und 2.2 folgende Grundsätze, die als Voraussetzung für das ressourcenschonende Bauen im Bestand Gültigkeit besitzen, extrahieren:

- Der Stoffdurchsatz im Bestand muss verlangsamt werden. Gebäude und Bauteile müssen weiterverwendet werden.
- Sanierungen müssen mit einfachen Mitteln, unter Verwendung reparaturfähiger Konstruktionen, erfolgen. Das Einbringen neuer Stoffe in den Lebenszyklus ist zu vermeiden.

- Die ästhetische Modernisierung des Bestandes muss durch das Gebot materieller Weiternutzung reduziert werden.
- Der Bestand muss mit seinem Potential als Angebot und nicht als Last gesehen werden. Bestandsbauten können positive Definitionselemente historischer Orte sein [Jessen und Schneider, 2000b].
- Das Arbeiten mit der Bausubstanz statt gegen die Bausubstanz ist Voraussetzung für die Reduzierung der Kosten einer Revitalisierungsmaßnahme [Kirchhoff, Jacobs et al., 1996].

Diese Grundsätze haben Auswirkungen auf den Lebenszyklus der Bestandsbauten und beeinflussen entscheidend die planerische Vorgehensweise bei einer Revitalisierungsmaßnahme:

- Bestandsbauten müssen intelligent und flexibel weitergenutzt werden.

Folgende Alternativstrategien zum Neubau verlängern den Lebenszyklus der Gebäude:

- **Das Umbauen** durch stufenweise, strukturelle und/oder organisatorische Dekomposition:
  - mit geringen baulichen Eingriffen
  - mit erheblichen baulichen Eingriffen
  - mit umfangreichen baulichen Eingriffen
- **Das Umnutzen** durch Tauschen von Flächen oder Gebäuden:
  - ohne bauliche Eingriffe (organisatorisch)
- **Der Bauverzicht** durch Nutzung von IuK-Technologien

Anwendung finden diese Grundsätze auf den heterogenen Gebäudebestand aus den Jahren 1871-1990. Dieser umfasst im Wesentlichen die folgenden Gebäudetypologien [Kohler, 1999]:

- Handels- und Lagergebäude
- Fabrik- und Werkstattgebäude
- landwirtschaftliche Betriebsgebäude
- sonstige Nichtwohngebäude

Der strukturelle Aufbau dieser Gebäude ist von besonderer Bedeutung für die konzeptionelle Vorgehensweise bei einer Revitalisierungsmaßnahme. Relevant sind hier vor allem große Gebäude mit den folgenden Eigenschaften:

- **Gebäude mit nicht trennbaren Primär-, Sekundär- und Tertiärstrukturen**, die vornehmlich in Massivbauweise erstellt worden sind (insbesondere aus dem Zeitraum von 1871-1948). Dieser Gebäudetyp wird nachfolgend **Typ A** genannt.
- **Gebäude mit trennbaren Primär-, Sekundär- und Tertiärstrukturen**, die vornehmlich in System- oder Skelettbauweise erstellt worden sind (insbesondere aus dem Zeitraum von 1949-1990). Dieser Gebäudetyp wird nachfolgend **Typ B** genannt.

Kapitel 2.3 verdeutlicht, dass es keine allgemeingültigen Entwurfs- oder Planungsmethoden gibt, die in der Praxis angewendet werden. Es werden unterschiedliche Methoden und Techniken eingesetzt, die auch in Abhängigkeit von der jeweiligen Aufgabe neu definiert werden. Festzuhalten ist, dass:

- es einen übergeordneten Planungsprozess gibt, innerhalb dessen das Entwerfen als ergebnisorientierter, kontinuierlich zu aktualisierender Prozess definiert werden kann,
- Entwurfsergebnisse zu einem frühen Planungszeitpunkt vorliegen müssen,
- alternative Entwurfslösungen erforderlich sind und
- normative Faktoren Einfluss auf das Entwurfsergebnis nehmen.

Aus den beschriebenen Strategien können zwei grundlegende Modelle für die planerische Bearbeitung der Weiternutzung des Gebäudebestandes abgeleitet werden.

Das erste Modell akzeptiert die bauliche Veränderung der Gebäude und integriert Maßnahmen, welche die Veränderung, respektive die Zerstörung von Gebäudebestandteilen, in Kauf nehmen. Diese Maßnahmen sind die stufenweise durchführbaren, strukturellen und organisatorischen Umbauten des Gebäudes, die durch Veränderung von Bauteilen erreicht werden. Auf Grund dieser Herangehensweise wird dieses Modell nachfolgend als „Zerstörungsreiches Modell“ bezeichnet.

Das zweite Modell basiert auf der Prämisse, dass keine baulichen Veränderungen am Gebäudebestand durchgeführt werden. Durch die organisatorische Umstrukturierung von Flächen innerhalb des Gebäudes oder den Tausch von Flächen zwischen mehreren Gebäuden werden neue Nutzungsarten im Gebäudebestand ermöglicht. Da bei diesem Modell absehbar ist, dass eine vollständige Bedarfserfüllung im Abgleich mit dem Bestandsangebot selten zu erreichen ist, wird unter der Prämisse geringfügiger Eingriffe in die Sekundärstruktur, d.h. der Addition oder Subtraktion von Raumeinheiten, dieses Modell nachfolgend als „Zerstörungsarmes Modell“ bezeichnet. Eine genaue Betrachtung der aus der Anwendung dieser Modelle resultierenden Aufgaben in Kapitel 3 verdeutlicht, dass es sich bei der Lösung von so begriffenen Revitalisierungsaufgaben um eine Problemstellung handelt, die einem Umschichten, Umsortieren und Neuordnen von Räumen und Funktionen entspricht.

### 3 Modellbildung

Kapitel 3 beschreibt zwei Modelle zur Lösung von Aufgaben der Grundrissplanung in der Revitalisierung. Die Modelle demonstrieren die Vorgehensweise bei der Revitalisierung der exemplarisch gewählten Gebäudestrukturen der Typen A und B, die aus der Kategorisierung des Gebäudebestandes hervorgegangen sind (vgl. Kap. 2.4). Die Modelle sind unabhängig von der Verwendung digitaler Hilfsmittel. Die daraus resultierenden Anforderungen dienen als Grundlage für die Spezifikation informationstechnischer Werkzeuge, mit deren Hilfe die Modelle in eine Systemkonzeption überführt werden (vgl. Kap. 5). Die Modellbildung unterscheidet zwei unterschiedliche Vorgehensweisen, respektive Modelltypen.

**Zerstörungsreiche Modelle** bilden die Umbaustrategie ab. Große Freiräume innerhalb einer vorgegebenen Grundrisskonstellation werden mit Räumen des Raumprogramms gefüllt und auf diese Weise neu geordnet. Dabei wird der Abbruch von Wänden und Bauteilen in Kauf genommen. Dieses Modell wird hauptsächlich auf Gebäude des Typs B (trennbare Primär-, Sekundär- und Tertiärstrukturen), die vornehmlich in System- oder Skelettbauweise erstellt worden sind, angewendet.

**Zerstörungsarme Modelle** bilden die Umnutzungsstrategie ab, d.h. dem Versuch, die existente Raumstruktur eines Gebäudes einer neuen Nutzung zuzuführen und auf den Abbruch von Wänden und Bauteilen weitestgehend zu verzichten. Wesentliches Kriterium für dieses Modell sind die zwischen den Raumeinheiten existierenden Strukturen, die auf Übereinstimmung mit den Strukturen einer projektierten Nachnutzung untersucht werden. Dieses Modell wird hauptsächlich auf Gebäude des Typs A (nicht trennbare Primär-, Sekundär- und Tertiärstrukturen), die vornehmlich in Massivbauweise erstellt worden sind, angewendet.

Die Aufgaben bei der Anwendung Zerstörungsreicher und Zerstörungsarmer Modelle im Planungsprozess sind anspruchsvoll. Das planerische Zerlegen eines Gebäudes in strukturelle oder organisatorische Einheiten ist ein Prozess, der beim Entwerfen von Neubauten alltäglich ist.

Anders als beim Neubauentwurf, unterliegt das Umschichten, Umsortieren und Neuordnen von existenten Gebäudeeinheiten (Räumen und Raumgruppen) jedoch der Beschränkung, dass die Bauteile des zu revitalisierenden Gebäudes nicht oder nur geringfügig verändert werden sollten, um die existente Gebäudestruktur weitestgehend zu erhalten. In der Praxis werden beide Modelle gleichermaßen Anwendung finden, da absehbar ist, dass eine Umnutzung nie vollständig die durch den Bestand gegebenen Voraussetzungen erfüllen kann. Es ist daher erstrebenswert, zuerst mit Hilfe des Zerstörungsarmen Modells den Bestand auf Äquivalenz zum geforderten Raumprogramm zu untersuchen. Dabei entstehende nicht nutzbare Restflächen, können mit Hilfe des Zerstörungsreichen Modells neu beplant werden. Beide Modelle basieren auf einer Beschreibung von Zusammenhängen zwischen den Räumen des zu realisierenden Raumprogramms und den existenten Räumen des zu revitalisierenden Gebäudes. Die Modelle beschreiben exemplarisch, welche Parameter für die Umsetzung in der Systemkonzeption Relevanz besitzen.

### 3.1 Revitalisierungsmodelle

Beide Modelle adaptieren eine gängige Vorgehensweise von Architekten bei der Bearbeitung einer Revitalisierungsmaßnahme, die nachfolgend beschrieben wird. Dabei wird der Gebäudegrundriss in logische Einheiten zerlegt, die anschließend unter Verwendung von Bedingungen neu geordnet werden. Diese Neuordnung geschieht im Allgemeinen mit dem Anspruch, eine funktionale Lösung zu erreichen. Als Grundlage dieser Neuordnung dient ein schriftliches Raumprogramm, welches die in einer Neuordnung zu berücksichtigenden Räume nach der Art der Nutzung unter Angabe der erforderlichen Flächengröße auflistet.

#### Dekomposition

Zu Beginn der Planung erfolgt die Dekomposition des Gebäudegrundrisses in dessen inhärente strukturelle oder organisatorische Bestandteile. Dekomposition bedeutet in diesem Fall, dass aus einem Grundriss durch Unterteilung oder Zerlegung kleinere logische Einheiten gebildet werden. Es ist jedoch ebenso möglich, dass ein Grundriss, welcher bereits aus vielen kleineren Einheiten besteht, dadurch strukturiert wird, dass diese Einheiten zu größeren Einheiten zusammengefasst werden (Komposition). So können z.B. aus einer Anzahl zusammengehöriger Flächen gleicher Nutzung, übergeordnete Flächen gebildet werden (z.B. HNF + NNF  $\rightarrow$  NF, gem. DIN 277). Die Tiefe der Dekomposition / Komposition ist abhängig von der Komplexität des jeweiligen Gebäudegrundrisses und kann innerhalb eines Grundrisses auch variieren. Bei komplexen Grundrissen mit einer Vielzahl von Räumen werden diese häufig zu größeren Einheiten zusammengefasst (Bottom-Up). Nach der ersten Neuordnung werden die größeren Einheiten wiederum in einzelne, kleinere Einheiten zerlegt (Top-Down). Weniger komplexe Grundrisse werden in aller Regel durch die Veränderung der Anordnung oder der Größe einzelner Flächen neu geordnet. Beide Prozesse sind jedoch nicht voneinander zu trennen, da sowohl die Bildung größerer Raumeinheiten, als auch die Bearbeitung einzelner Räume in nahezu jeder Grundrisskonstellation auftritt. Abb. 3.1 verdeutlicht die strukturelle, stufenweise Dekomposition des existenten Gebäudes von der Tertiär- bis zur Primärstruktur eines Gebäudes vom Typ B.

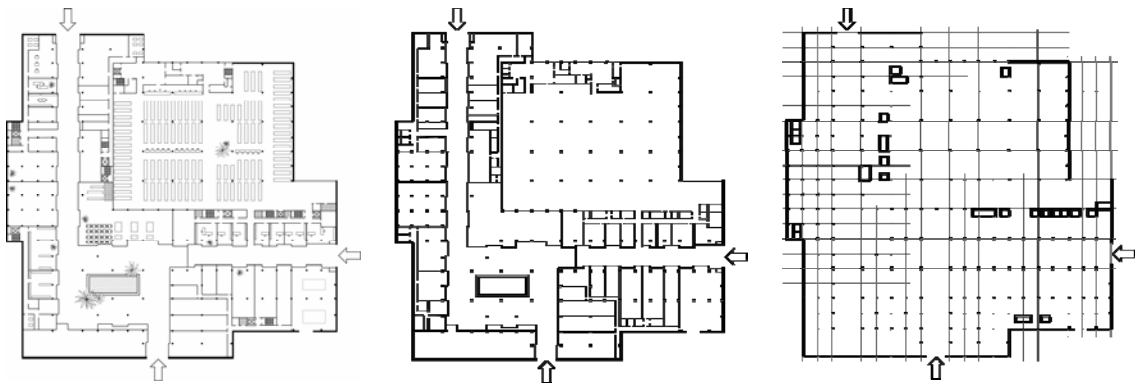


Abb. 3.1 Stufenweise, strukturelle Dekomposition um Tertiär- und Sekundär- zur Primärstruktur

Abb. 3.2 veranschaulicht die schrittweise organisatorische Dekomposition des Gebäudegrundrisses in Anlehnung an die Klassifikation der Flächen nach DIN 277 T1 und T2.

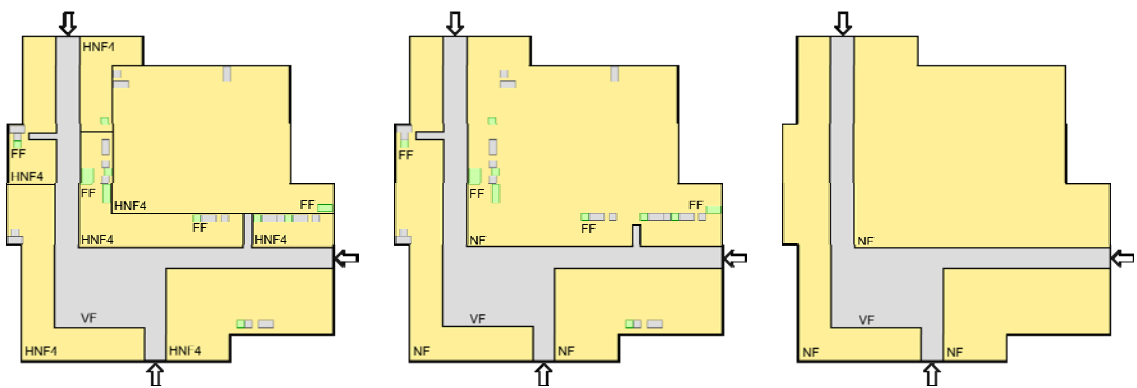


Abb. 3.2 Organisatorische Dekomposition – Einzelflächen / Nutzungseinheiten / Gesamtflächen

### Neuordnung

Um eine Neuordnung der Flächen innerhalb einer gegebenen Grundrisskonstellation zu erreichen, werden die Eigenschaften der Flächen verändert. Diese Eigenschaften können numerischer (z.B. Position) und / oder alphanumerischer Art (z.B. Nutzungsart) sein.

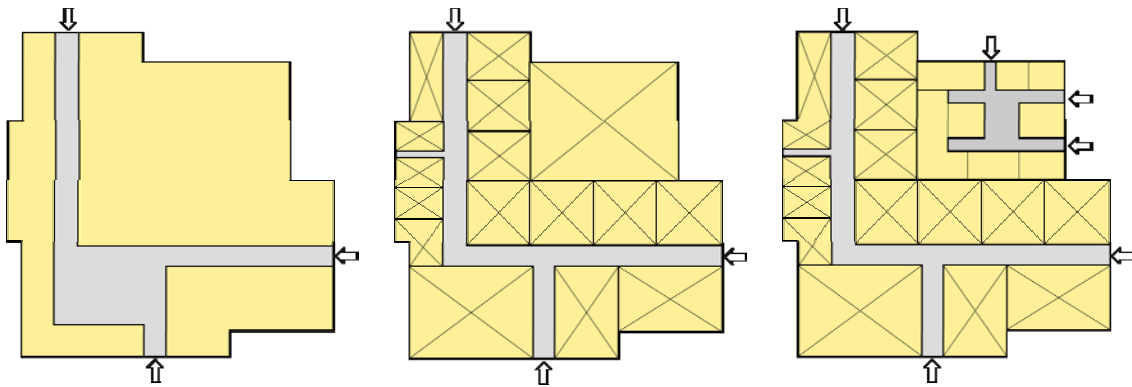


Abb. 3.3 Stufenweise Neuordnung einzelner Flächen aus Abb. 3.2

### Bedingungen

Die Neuordnung der Räume geschieht in der Regel nicht zufällig, sondern unterliegt bestimmten Bedingungen. Diese Bedingungen können z.B. funktionaler, konstruktiver oder gestalterischer Natur sein. Wesentlich ist, dass der Architekt diese Bedingungen beim Entwerfen, unterstützt durch das manuelle Skizzieren, bewusst aber auch unbewusst einsetzt und dadurch Beziehungen zwischen Räumen oder Raumeinheiten herstellt.

### Ziele

Das Entwerfen und Verwerfen von Linien beim manuellen Skizzieren, das eine Änderung der Eigenschaften eines Raumes nach sich zieht, geschieht nicht allein unter der Prämisse, einen Raum zu skalieren oder zu positionieren, sondern dient der Verfolgung eines übergeordneten, jedoch in der Regel nicht quantifizierbaren Zieles. Dieses übergeordnete Ziel ist die Herstellung einer funktionalen und ästhetischen Gesamtanordnung der Einzelelemente zu einer harmonischen Gesamtkonzeption. Das „Herantasten“ an eine Entwurfslösung durch das manuelle Skizzieren ist dabei ein nicht zu unterschätzendes, jedoch auch nicht zu formalisierendes Element der Entwurfsfindung, da es keinen festgelegten Regeln folgt. Dennoch existieren bei der Entwurfsbearbeitung untergeordnete Teilziele, die quantifizierbar sind. Ein derartiges Teilziel ist z.B. die Erfüllung von Anforderungen des Raumprogramms.

### Gemeinsamkeiten der Modelle

Beiden Modellen ist die Vorgehensweise der Dekomposition des Grundrisses, der Definition von Bedingungen zwecks Erreichung eines übergeordneten Zieles und der anschließenden Neuordnung der Räume gemeinsam.

Die Modelle basieren auf der Veränderung der Eigenschaften von Flächen. Eine Fläche kann einen einzelnen Raum (z.B. Wohnraum) oder auch mehrere Räume in Anlehnung an eine bestimmte Konvention (z.B. HNF 4 Lagerräume, gem. DIN 277 T2) abbilden. Der Begriff „Flächeneinheit“ bezeichnet organisatorisch zusammengehörige Räume (z.B. Laden, incl. Verkaufsf-

raum, Personalraum, Toiletten). Die Eigenschaften der Flächen können sowohl variable als auch konstante Werte annehmen.

### Dekomposition

- Eine Fläche kann einen einzelnen Raum oder mehrere Räume abbilden.
- Eine Fläche kann eine Funktion abbilden (z.B. HNF 1, Wohnen, gem. DIN 277 T2).
- Eine Flächeneinheit besteht aus mehreren organisatorisch zusammengehörigen Flächen
- Flächenproportionen werden durch die Maße logischer Einheiten definiert (z.B. Mauerwerksmaße, Rastermaße).
- Es gibt Flächen, deren Proportionen keinem Maß einer logischen Einheit folgen (z.B. durch Einbau von Trennwänden).
- Mehrere kleine Flächen können zu einer größeren Fläche zusammengesetzt werden (Bottom-Up).
- Eine größere Fläche kann in mehrere kleinere Flächen zerlegt werden (Top-Down).

### Bedingungen

- Beziehungen zwischen den Flächen definieren deren Neuordnung. Diese Beziehungen können normativen Charakter haben.

### Ziele

- Quantifizierbare Teilzieldefinitionen bestimmen die Neuordnung der Flächen.
- Quantifizierbare übergeordnete Entwurfsziele werden in der Regel nicht definiert.
- Quantifizierbare übergeordnete Entwurfsziele können aus der Summe zu gewichtender quantifizierbarer Teilziele gebildet werden.
- Die Neuordnung von Flächen des Raumprogramms kann, unter Erfüllung funktionaler, konstruktiver und ästhetischer Rahmenbedingungen, als Lösung einer Entwurfsaufgabe und somit als Erfüllung eines Entwurfszieles verstanden werden (Constraint Satisfaction).

### Neuordnung

- Die Neuordnung einer Fläche kann durch Veränderung ihrer Eigenschaften (Größe, Position, Funktion) erfolgen.
- Die Vergrößerung einer Fläche kann entweder durch Skalieren oder durch Addition mit einer anderen Fläche erfolgen.
- Die Verkleinerung einer Fläche kann entweder durch Skalieren oder durch Subtraktion einer anderen Fläche erfolgen.



## 3.2 Zerstörungsreiches Modell

### 3.2.1 Methodik

Das Zerstörungsreiche Modell basiert auf den inhärenten Strukturen eines existenten Gebäudes, welche aus der strukturellen oder organisatorischen Dekomposition des Gebäudegrundrisses hervorgegangen sind. Diese vorgefundenen Strukturen werden als Grundlage der Neuplanung verwendet und zur Lösungsfindung eingesetzt. Beim Zerstörungsreichen Modell sind diese Strukturen große leere Flächen, die mit den im Raumprogramm aufgelisteten Räumen gefüllt werden. Sofern diese freien Flächen nicht bereits vorhanden sind, werden sie durch Dekomposition vom Architekten erzeugt, d.h. der späteren Verwendung des Zerstörungsreichen Modells geht die bauliche Veränderung der zu beplanenden Geschossfläche voraus. Das Ziel dieses Modells ist es, die im Raumprogramm definierten Flächen mit Hilfe der durch den Architekten zu definierenden Beziehungen, innerhalb dieser freien Flächen anzuordnen.

#### Dekomposition

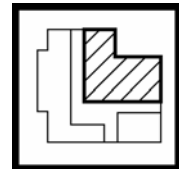
Dekomposition

Zerlegung des Gebäudegrundrisses in strukturelle oder organisatorische Flächen. Beispiel: Hauptnutzflächen und Verkehrsflächen (gem. DIN 277 T2).



Auswahl

Definition einer Fläche als neu zu beplanende Grundfläche. Beispiel: Auswahl der Hauptnutzfläche HNF4 eines Einkaufszentrums.



#### Bedingungen

Raumprogramm

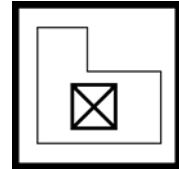
Festlegung des Raumprogramms. Beispiel: Planung von HNF4.1 Lager-, 4.3 Kühl-, 4.5 Verkaufsräumen.



**Beziehungen zwischen den anzuordnenden Flächen und der Grundfläche:**

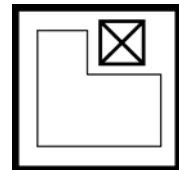
Innerhalb

Flächen müssen innerhalb der Grundfläche liegen. Beispiel: Alle Flächen HNF4.1, 4.3 und 4.5 innerhalb der Hauptnutzfläche HNF 4.



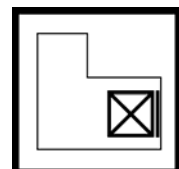
Außerhalb

Flächen müssen (bei Nichterfüllung eines Teilzieles) außerhalb der Grundfläche liegen. Beispiel: HNF4.1 Lager liegt außerhalb HNF4.



Außenseite

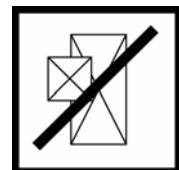
Flächen müssen auf einer Außenseite der Grundfläche liegen. Beispiel: Die Fläche eines Verkaufsräumens HNF 4.5 mit Schaufenster muss an einer Außenseite liegen.



**Beziehungen zwischen den anzuordnenden Flächen:**

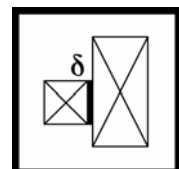
Keine Überschneidungen

Flächen dürfen sich nicht überschneiden. Beispiel: Nutzfläche NF und Verkehrsfläche VF.



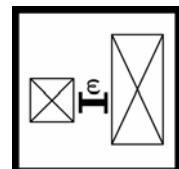
Berührung

Flächen müssen sich berühren. Beispiel: Die Fläche eines Kühlraumes HNF 4.3 muss an die Verkaufsfläche HNF 4.5 für Frischwaren angrenzen.



Abstand

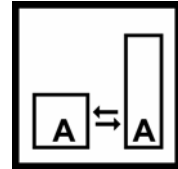
Zwischen Flächen muss ein Abstand bestehen. Beispiel: Die Fläche eines Fluchttreppenraumes darf nicht direkt neben der Fläche eines notwendigen Treppenraumes liegen.



**Eigenschaften der anzuordnenden Flächen:**

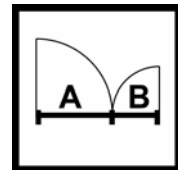
Flächenmaße

Eigenschaften der jeweiligen Flächen. Beispiel: Definition der Flächenmaße, wie Länge, Breite, Flächeninhalt, Umfang einer Fläche.



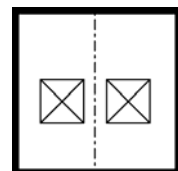
Proportionen

Beziehungen der Teilmaße von Flächen. Beispiel: Verhältnis von Länge / Breite, um eine bestimmte Flächenproportion zu erhalten.



Anordnungen

Beziehungen zwischen mehreren Flächen. Beispiel: Symmetrische Anordnung von zwei Verkaufsflächen HNF 4.5 in Bezug zur Mittelachse einer Verkehrsfläche VF.



**Ziele**

Minimierung / Maximierung

Definition von Zielen / Teilzielen. Beispiel: Minimierung oder Maximierung der Größe von Flächeneinheiten (minimale Verkehrsfläche, maximale Nutzfläche).



Constraint Satisfaction

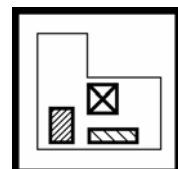
Definition von Zielen / Teilzielen. Beispiel: Erfüllung aller Bedingungen als Zielvorgabe (Constraint Satisfaction), wenn keine Teilziele definiert werden können.



**Neuordnung**

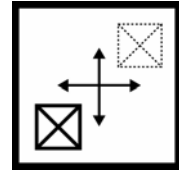
Neuordnung

Anordnung der Flächen des Raumprogramms innerhalb der Grundfläche. Beispiel: Alle Flächen HNF4.1, 4.3 und 4.5 müssen die vorgenannten Bedingungen erfüllen.

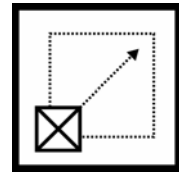


**Positionieren**

Neuordnung der Flächen durch Änderung der Position. Beispiel: Positionsänderung durch Änderung der Ursprungskoordinaten

**Skalieren**

Neuordnung der Flächen durch Änderung der Proportion. Beispiel: Proportionsänderung durch Änderung von Länge und / oder Breite.

**Ergebnis****Keine Lösung**

Beispiel: Es existiert keine Lösung, welche die definierten Bedingungen und Ziele, durch Änderung der Positionen und / oder Proportionen der Flächen des Raumprogramms, erfüllt.

**Lösung**

Beispiel: Es existiert mindestens eine Lösung, welche die definierten Bedingungen und Ziele, durch Änderung der Positionen und / oder Proportionen der Flächen des Raumprogramms, erfüllt.

**Zeitlicher Ablauf**

Das Entwerfen ist ein kontinuierlich zu aktualisierender, ergebnisorientierter Prozess (vgl. Kap. 2.3). Der zeitliche Ablauf dieses Prozesses folgt beim Zerstörungsreichen Modell einer klar definierten Vorgehensweise. Der Architekt beginnt zunächst mit der Zerlegung des Gebäudegrundrisses in größere logisch zusammenhängende Flächen, die unabhängig voneinander beplant werden können. Die jeweilige zu beplanende Fläche wird anschließend mit den Räumen eines existenten Raumprogramms, welches die Anzahl der Räume mit den Angaben über die Art der Nutzung und die Flächengrößen beinhaltet, gefüllt. Diese Räume, die im Modell lediglich wieder Flächen darstellen, werden mit Bedingungen versehen, die ihre Neuordnung definieren. Diese Bedingungen können obligatorisch (z.B.: alle Räume befinden sich innerhalb der zu beplanenden Fläche) oder fakultativ (Raum A berührt Raum B) sein. Die Neuordnung wird unter der Prämisse durchgeführt, ein Ziel zu erreichen. Diese Zieldefinition erfolgt im Anschluss an die Vergabe der Bedingungen. Auch hier gibt es obligatorische und fakultative Ziele. Ein fakultatives Ziel kann z.B. die Erzeugung einer möglichst kleinen Verkehrsfläche (Minimierung) oder einer möglichst großen Nutzfläche (Maximierung) sein. Auch die Kombination beider Ziele ist durchaus erstrebenswert, d.h. aus zwei Zielen werden Teilziele gebildet, deren Summe ein übergeordnetes Gesamtziel darstellt. Dieses Gesamtziel ist eindeutig

quantifizierbar. Da der Architekt jedoch in der Regel keine quantifizierbaren Gesamtziele definiert, muss es möglich sein, ohne Zieldefinition zu einer Lösung zu gelangen. Da die Neuordnung der Flächen an Bedingungen gebunden ist, die in jedem Fall erfüllt werden müssen, kann die Erfüllung dieser Bedingungen (Constraints) als obligatorische Zieldefinition angesehen werden (Constraint Satisfaction). Der Zieldefinition folgt die Neuordnung der Flächen durch Positions- oder Größenveränderungen. Bei Erreichen des Zieles werden eine oder mehrere Lösungen erzeugt. Wird das Ziel nicht erreicht, können die Zieldefinitionen, die Bedingungen, die Dekomposition und unter Umständen auch das Raumprogramm verändert werden. Mit den iterativ veränderten Werten erfolgen wiederum kontinuierliche Versuche, die Flächen neu zu ordnen. Diese Vorgehensweise kann ebenfalls verwendet werden, um alternative Lösungsmöglichkeiten einer Entwurfsaufgabe nach gleichen oder grundsätzlich verschiedenen Anforderungen zu erzeugen (vgl. Kap. 2.3). Die Integration geometrischer Bedingungen ist ebenso möglich, wie die Verwendung normativer Bedingungen.

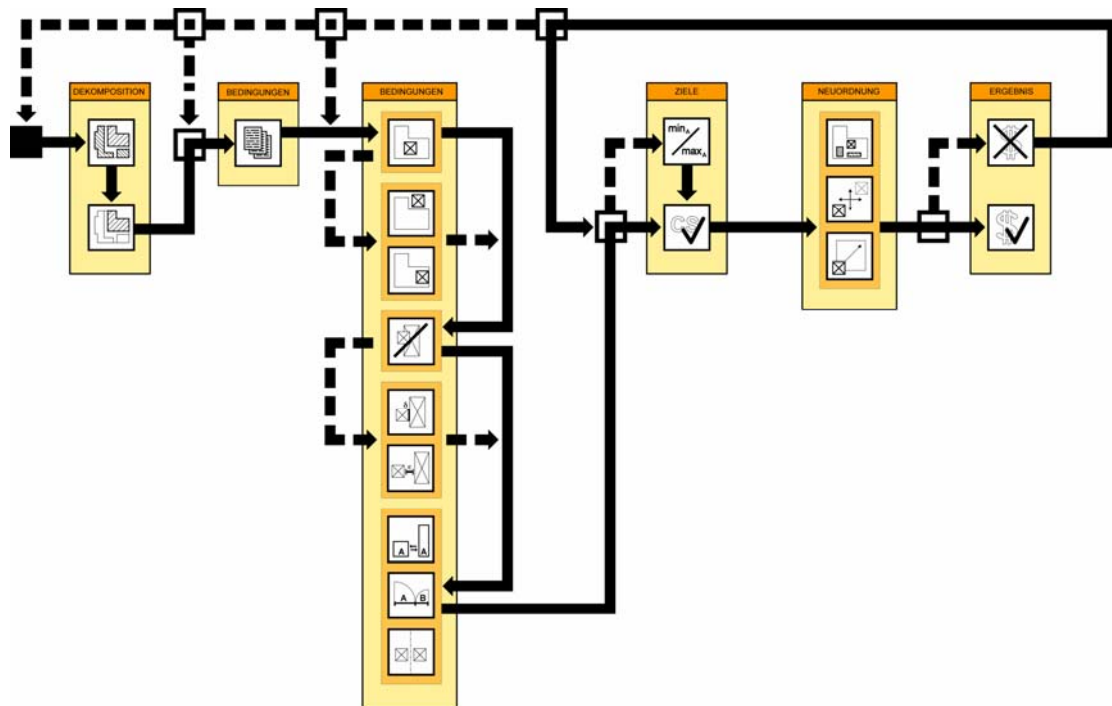


Abb. 3.4 Zeitlicher Ablaufplan des Zerstörungsreichen Modells

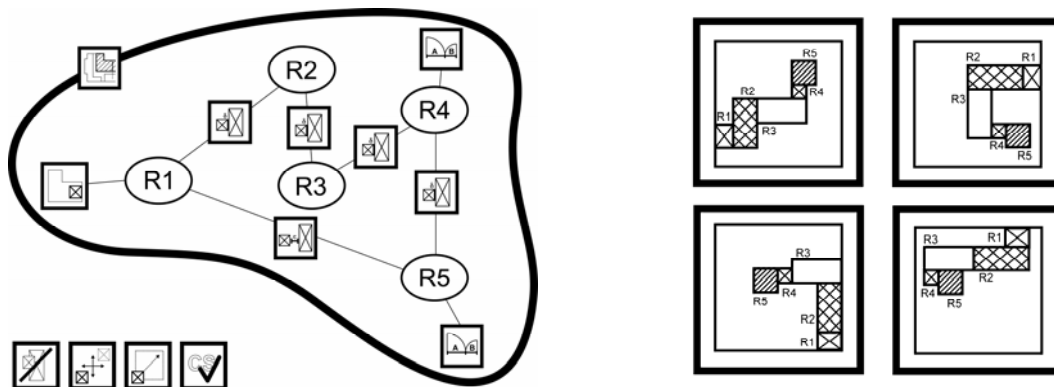


Abb. 3.5 Entwurfsaufgabe (Zerstörungsreiches Modell) - Alternative Lösungen nach gleichen Anforderungen

### 3.2.2 Interdisziplinäre Analogien der Methodik

Die Methodik des Zerstörungsreichen Modells beruht auf der schrittweisen Zerlegung eines Gebäudes in kleinere Flächen. Die sich daran anschließende, in der Regel differente Komposition weiterer Flächen zu einer neuen Anordnung, findet sich auch in anderen Disziplinen, in denen die Methodik des Sortierens, Umschichtens oder Verlagerens zur Erfüllung eines Zieles eingesetzt wird. Zum besseren Verständnis der Thematik und zur Ableitung von Problemlösungsstrategien werden einige Anwendungen hier kurz vorgestellt.

Ein einfaches Beispiel sind die Warenregale in Supermärkten. In diesen Regalen befinden sich Plätze, die der temporären Lagerung von Lebensmitteln, Haushaltswaren oder anderen Gebrauchsgütern dienen. Das Platzangebot ist in der Regel begrenzt, und nur bestimmte Waren können in bestimmte Regale einsortiert werden. Die Größe des Regals, der Standort innerhalb des Supermarktes und die Benachbarung zu anderen Regalen definiert das Warenangebot, welches im Regal platziert wird. Auch die inhärenten Eigenschaften der Ware haben Einfluss auf den Standort. Verderbliche Waren werden so positioniert, dass sie relativ schnell vom Kunden gegriffen werden können, Kühlwaren werden in Kühlregale einsortiert, Waren mit hohem Gewicht in Schwerlastregale. Alle Waren einer bestimmten Warengruppe finden sich gruppiert zu einer logischen Einheit, die wiederum in Beziehung zu anderen logischen Einheiten steht. Ähnliche, jedoch weitaus komplexere Modelle finden sich im Bereich der Logistik. Ein Containerterminal besteht aus einer klar abgegrenzten Fläche zur Lagerung von Containern, die so positioniert werden müssen, dass deren Verladung, in Abhängigkeit von zeitlichen Bedingungen, mit möglichst geringem Aufwand realisiert werden kann. Die Container sind von genormter, jedoch unterschiedlicher Größe. Zwischen den Lagerflächen befinden sich Transportwege, welche die Erreichbarkeit der Container zur Lagerung oder Verladung gewährleisten. Die Analogie in der Architektur sind Raumeinheiten, welche aus einer bestimmten Anordnung von Räumen bestehen und durch Verkehrsflächen erschlossen werden.



Abb. 3.6 CTA Containerterminal, Hamburg Altenwerder, Analogie der Modellbildung

Modelle, welche die Lagerung oder den Transport von Waren beschreiben, werden im Wissenschaftsgebiet des Operations Research als Probleme bezeichnet. Bei der hier vorliegenden Problemkategorie handelt es sich (unter Vernachlässigung von Transportproblemen) um so genannte Verpackungsprobleme (engl. Packing- oder BinPacking-Problems). Diese Problemkategorie kann am Beispiel des „Perfect Square Problems“ verdeutlicht werden. Das Problem besteht darin, eine Anzahl  $n$  Quadrate mit vorgegebenen Seitenlängen innerhalb eines größeren Quadrates so anzuordnen, dass die gesamte Fläche des großen Quadrates durch die Summe der Quadrate  $n$  so ausgefüllt wird, dass keine Leerräume verbleiben. Dabei dürfen sich die Quadrate nicht überschneiden.

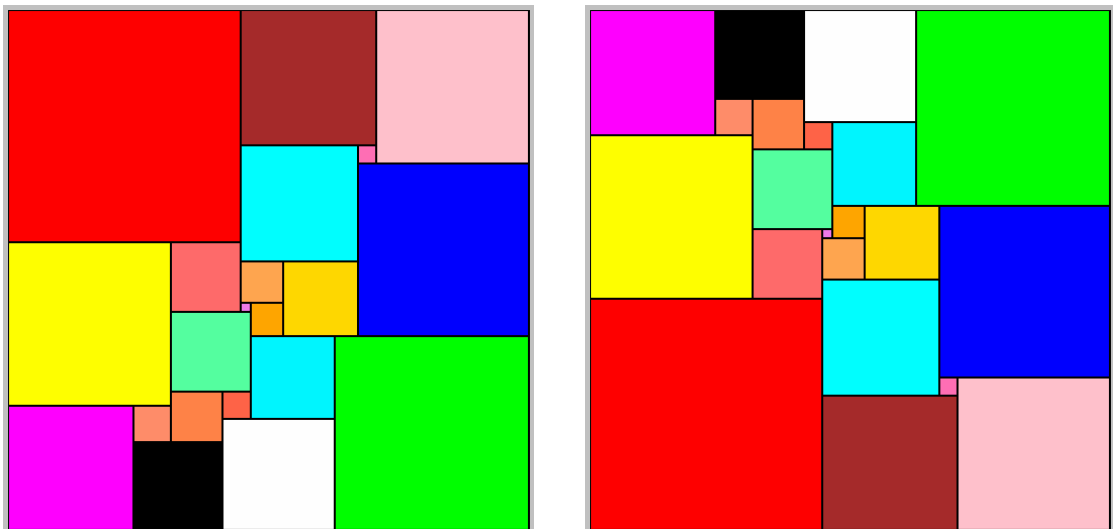


Abb. 3.7 Unterschiedliche Lösungen eines Perfect Square Problems mit 21 Quadraten

Das Zerstörungsreiche Modell besitzt methodische Analogien zu den Verpackungsproblemen. Im Unterschied zum Perfect Square Problem, gibt es jedoch keine Klasse von Objekten die durch feste Größen definiert sind. Ferner unterliegt die Anordnung der Räume zusätzlichen Bedingungen (Position, Proportion, Raumbeziehungen), welche die Lösung des Problems ungleich aufwändiger gestalten.

### **Kurzzusammenfassung**

Das Zerstörungsreiche Modell wird auf große Flächen innerhalb des Gebäudegrundrisses angewendet, die mit den variablen Flächen des Raumprogramms gefüllt werden müssen. Die Bedingungen, wie diese Flächen angeordnet werden müssen, sind nicht bekannt und müssen vom Architekten definiert werden. Die Neuordnung geschieht durch Änderung der Positionen und / oder der Proportionen der Flächen.

## **3.3 Zerstörungsarmes Modell**

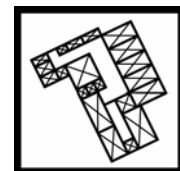
### **3.3.1 Methodik**

Das Zerstörungsarme Modell basiert ebenfalls auf den inhärenten Strukturen des existenten Gebäudes. Anders als beim Zerstörungsreichen Modell, werden diese jedoch ausschließlich aus der organisatorischen Dekomposition des Gebäudes in einzelne kleine Flächen gewonnen. Diese Flächen werden aus existenten Räumen gebildet. Es ist möglich, jedoch nicht wünschenswert, das Modell auf noch kleinere Einheiten anzuwenden, z.B. auf Flächen, die auf der Grundlage eines Rastermaßes zerlegt werden. Das Ziel dieses Modells ist es, im existenten Gebäudegrundriss Flächen zu finden, welche den Anforderungen des Raumprogramms entsprechen. Dies geschieht durch einen Vergleich der Eigenschaften der vorhandenen Flächen des Bestandsgrundrisses, mit den Eigenschaften der Flächen des Raumprogramms. Am bestehenden Gebäude werden dabei keinerlei bauliche Veränderungen vorgenommen. Dies ist möglich, da das Zerstörungsarme Modell nicht die geometrische Ausprägung eines Raumes, sondern lediglich die ihm zugehörige Nutzungsart verändert.

### **Dekomposition**

Dekomposition

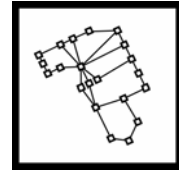
Zerlegung des Gebäudes in kleinere Flächen. Beispiel: Diese Flächen entsprechen den einzelnen Räumen des Gebäudes oder kleinen Nutzungseinheiten (z.B. Nasszellen).





**Benachbarung**

Beschreibung existenter Nachbarschaftsbeziehungen zwischen diesen Flächen. Beispiel: Flächen berühren sich oder liegen sich unmittelbar an einer Verkehrsfläche gegenüber. Diese Beziehungen werden als Benachbarung definiert.

**Bedingungen****Raumprogramm**

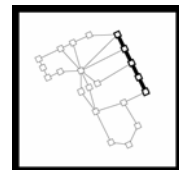
Festlegung des Raumprogramms. Beispiel: Die Flächen des Raumprogramms (Bürogebäude) bilden eine Flächeneinheit.

**Unterteilung des Raumprogramms**

Beispiel: Organisatorisch zusammengehörige Flächen werden aus dem Raumprogramm extrahiert, zu Flächeneinheiten gruppiert (Büronutzung A, Büronutzung B, Service-Bereich) und mit Eigenschaften (Größe der Einzelflächen, Gesamtgröße der Flächen, Anzahl der Flächen / Flächeneinheit) versehen.

**Restriktionen**

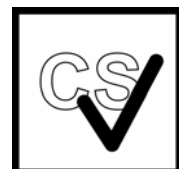
Festlegung der Restriktionen von Flächen einer zusammengehörigen Flächeneinheit. Beispiel: Jede Fläche einer Flächeneinheit muss mindestens eine weitere Fläche dieser Flächeneinheit berühren.

**Ziele****Minimierung / Maximierung**

Definition von Zielen / Teilzielen. Beispiel: Maximierung der gewichteten Summe der Erfüllung der Teilziele, d.h. annähernd erfüllte Ziele werden in einem übergeordneten Ziel gewichtet (z.B. 80 statt 85qm gesuchte Fläche einer Flächeneinheit).

**Constraint Satisfaction**

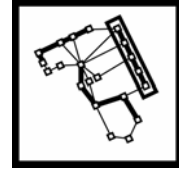
Definition von Zielen / Teilzielen. Beispiel: Es wird die Übereinstimmung der Eigenschaften zwischen den Flächeneinheiten des Raumprogramms mit den vorhandenen Flächen des Bestandsgrundrisses angestrebt.



## Neuordnung

### Neuordnung

Vergleich der Eigenschaften der Flächeneinheiten des Raumprogramms mit den vorhandenen Flächen des Gebäudegrundrisses. Beispiel: Es werden Flächen im Gebäudegrundriss gesucht, welche mit den Eigenschaften der Flächeneinheiten des Raumprogramms (Größe der Einzelflächen, Gesamtgröße der Flächen, Anzahl der Flächen / Flächeneinheit) übereinstimmen.



## Ergebnis

### Keine Lösung

Beispiel: Es existiert keine Lösung, welche die definierten Bedingungen und Ziele, erfüllt, da keine Übereinstimmung der Flächen des Gebäudegrundrisses mit den Eigenschaften der Flächeneinheiten des Raumprogramms gefunden werden konnte.



### Lösung

Beispiel: Es existiert mindestens eine Lösung, welche die definierten Bedingungen und Ziele, erfüllt, da eine Übereinstimmung der Flächen des Gebäudegrundrisses mit den Eigenschaften der Flächeneinheiten des Raumprogramms gefunden werden konnte.



## Zeitlicher Ablauf

Der zeitliche Ablauf des Zerstörungsfreien Modells ist weniger komplex als jener des Zerstörungsfreien Modells, da hier keine Bedingungen auf einzelne Flächen angewendet werden müssen. Der Architekt beginnt mit der Zerlegung des Gebäudegrundrisses in Flächen, die den vorhandenen Räumen entsprechen. Diese Vorgehensweise hat zur Folge, dass bei Erfüllung der späteren Zieldefinition eine Umnutzung ohne bauliche Veränderung stattfinden kann. Ein gewichtiger Faktor für die Zielerfüllung ist die Definition der Beziehungen zwischen diesen Flächen. Die signifikanteste Beziehung ist dabei die Benachbarung der Flächen, um bei der Zielerfüllung sicher zu stellen, dass auch die Flächen einer neu zu bildenden Flächeneinheit benachbart sind. Eine derartige Benachbarung bedeutet, dass jede Fläche mindestens eine weitere Fläche als Nachbar besitzen muss. Das heißt, zwischen zwei Flächen einer Flächeneinheit muss „quasi“ eine Berührungskante existieren. (Anm.: Es gibt Ausnahmen, bei denen Flächen als benachbart angesehen werden können, obwohl keine Berührungskante existiert. Dies ist beispielsweise der Fall, wenn sich Flächen an einer Verkehrsfläche unmittelbar gegenüberliegen). Da die vollständige Einpassung eines Raumprogramms in ein vorhandenes Gebäude selten möglich ist, besteht die Option, das Raumprogramm in logische Flächeneinheiten zu zerlegen und Äquivalenzen zu diesen Flächeneinheiten innerhalb des Gebäudegrundrisses zu suchen. Auch bei dieser Zerlegung besteht nur eine geringe Chance, eine vollständige Übereinstimmung

mit dem Bestand zu finden. Aus diesem Grund kann die Übereinstimmung jeder Flächeneinheit mit Flächeneinheiten des Gebäudegrundrisses als Teilziel festgelegt werden. Aus den Teilzielen lässt sich eine gewichtete Summe bilden, deren Maximierung als übergeordnetes Ziel definiert werden kann. Diese Vorgehensweise führt zwar zu einer qualitativen Verschlechterung potentieller Ergebnisse, jedoch zu einer quantitativen Vergrößerung des Lösungsraumes. Eine weitere Möglichkeit der Beeinflussung des Ergebnisses ist die Anwendung von Bedingungen auf die Flächeneinheiten des Raumprogramms. So kann eine Flächeneinheit z.B. durch die Größe der Einzelflächen, die Gesamtgröße der Flächen sowie die Anzahl der Flächen / Flächeneinheit festgelegt werden. Durch Kombinationen dieser Festlegungen ist innerhalb der vorgegebenen Grundrisskonstellation eine größere Varianz möglich.

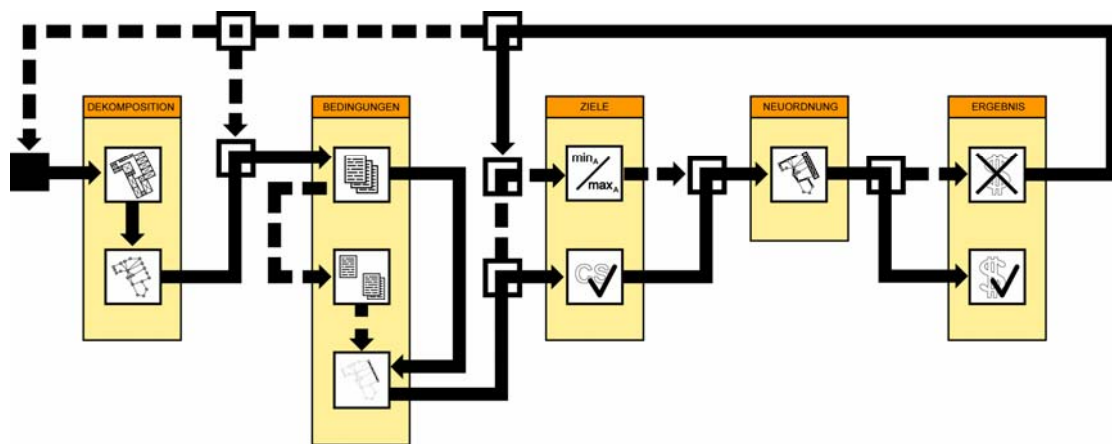


Abb. 3.8 Zeitlicher Ablaufplan des Zerstörungsarmen Modells

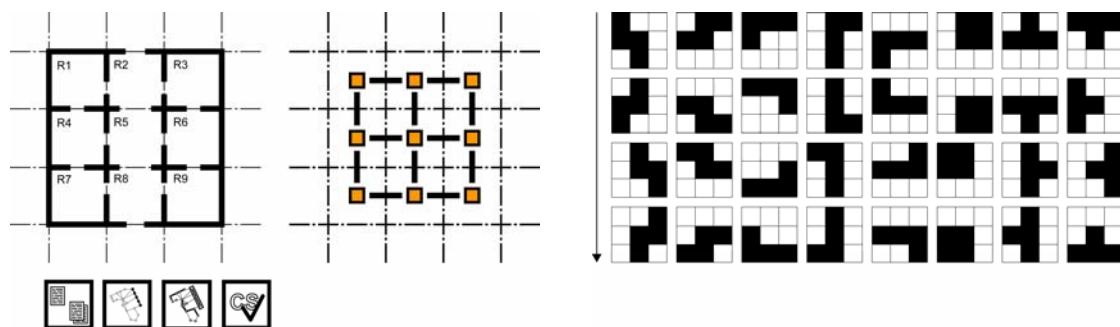


Abb. 3.9 Entwurfsaufgabe (Zerstörungsarmes Modell) - Alternative Lösungen nach gleichen Anforderungen

### 3.3.2 Interdisziplinäre Analogien der Methodik

Die Methodik des Zerstörungsarmen Modells ähnelt der Struktur von Brettspielen wie Schach, Go oder 4Gewinnt. Diese Spiele basieren auf einer Anzahl von Feldern oder Kreuzungspunkten von Rasterlinien, die in Abhängigkeit von Bedingungen (Spielregeln) mit Spielsteinen belegt werden. Die Zielerfüllung ist ebenso wie beim Zerstörungsarmen Modell klar definiert. Analog sind das festgelegte Raster, die eingeschränkte Möglichkeit Felder zu belegen und die Erfüllung eines übergeordneten Zieles, d.h. das jeweilige Spiel zu gewinnen.

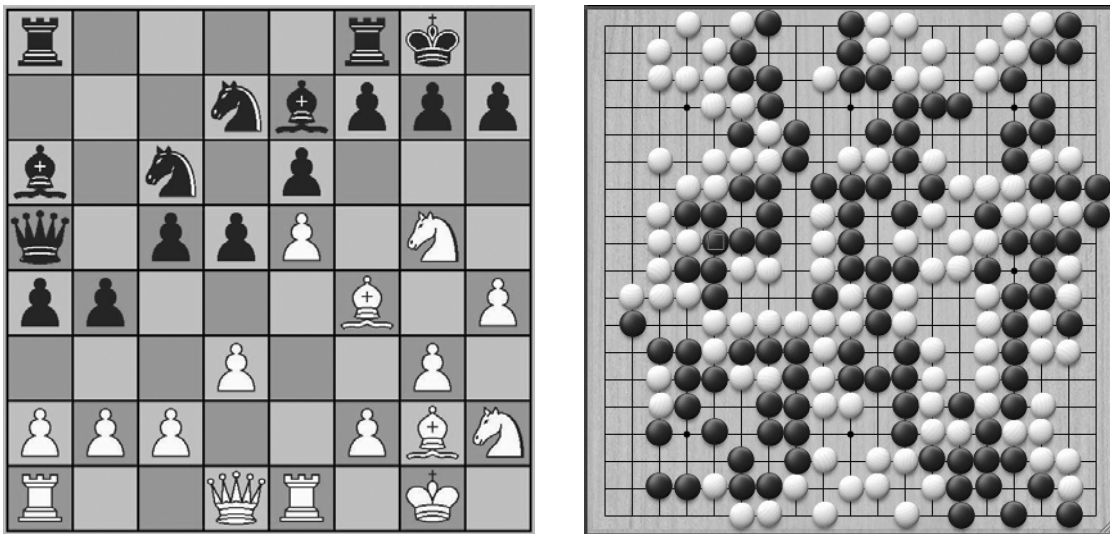


Abb. 3.10 Schach [hjb, 2004]; Go [Sente, 2005]

#### Kurzzusammenfassung

Das Zerstörungsarme Modell wird auf kleine Flächen innerhalb des Gebäudegrundrisses angewendet, die mit Flächen oder Flächeneinheiten des Raumprogramms verglichen werden. Die Bedingungen, die zwischen diesen existenten Flächen bestehen, sind bekannt. Die Neuordnung geschieht durch Änderung der Nutzungsart der Flächen.

### 3.4 Zusammenfassung und Implikationen

Aus der Methodik des Zerstörungsreichen Modells (nachfolgend ZR-Modell genannt) und des Zerstörungsarmen Modells (ZA-Modell) resultieren Fragen zu deren informationstechnischer Umsetzung. Beide Modelle arbeiten mit Repräsentationen von Räumen in Form von Flächen. Während das ZR-Modell die, auf dem Raumprogramm basierenden, bisher noch nicht existenten Flächen in Form geometrischer Objekte abbildet und verändert, arbeitet das ZA-Modell lediglich mit Nutzungseigenschaften vorhandener Flächen, unter Beibehaltung ihrer geometri-

schen Eigenschaften. Beide Modelle müssen Eigenschaften der Flächen verändern, um zu einer Lösung zu gelangen. Das ist einerseits die geometrische Ausprägung der Flächen des Raumprogramms (ZR-Modell) und andererseits die Zugehörigkeit der vorhandenen Flächen des Gebäudegrundrisses zu einer neuen, dem Raumprogramm entsprechenden, Flächeneinheit (ZA-Modell).

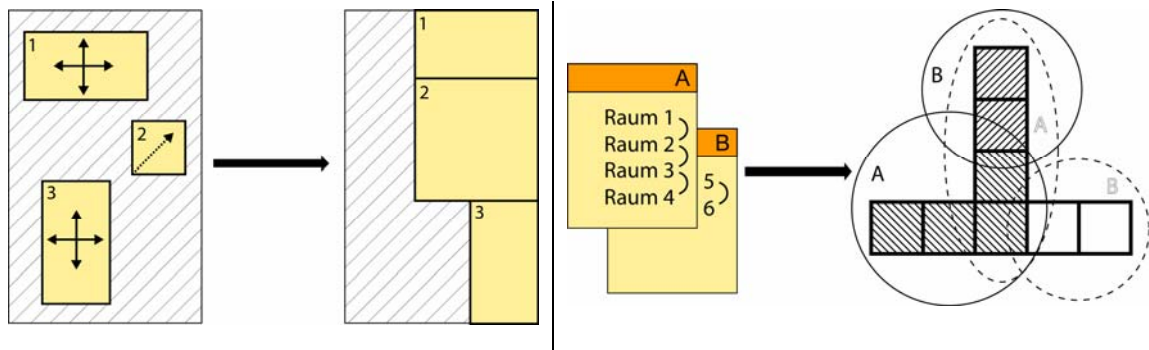


Abb. 3.11 Methodik des ZR-Modells, Methodik des ZA-Modells

Aus dieser Methodik ergibt sich eine der Kernfragen der Modellbildung:

*Welcher Gesetzmäßigkeit müssen Flächen folgen, um zu einer Neuordnung des Gebäudegrundrisses, die eine funktionale Lösung der Entwurfsaufgabe darstellt, zu gelangen?*

### 3.4.1 Regeln

Flächen können nur dann innerhalb eines Grundrisses funktional angeordnet werden, wenn das Wissen, wie man zu einer funktionalen Lösung gelangt und was eine funktionale Lösung ist, als Gesetzmäßigkeit, z.B. in Form von Regeln, festgelegt wird. Kapitel 2.3 verdeutlicht jedoch, dass es weder eine allgemeingültige Antwort auf die Frage gibt, nach welcher Methodik Architekten entwerfen, noch eine eindeutige Antwort auf die Frage, welche Gesetzmäßigkeiten sie anwenden, um Entscheidungen zu treffen, die sie zu funktionalen Lösungen gelangen lassen.

Die Planungspraxis und bestehende Gebäude zeigen jedoch, dass trotz teilweise komplexer Beziehungen zwischen den Flächen eines Grundrisses, Architekten zweifelsfrei in der Lage sind, diese funktional anzuordnen. Dies geschieht scheinbar ohne dass explizit nachgewiesen werden kann, welche Bedingung dazu geführt hat, dass sich eine Fläche an einer bestimmten Position befindet oder an eine andere Fläche angrenzt. Es ist zu vermuten, dass der Architekt beim Entwerfen Assoziationsketten bildet und so Zusammenhänge zwischen den anzuordnenden Flächen herstellt. Er kann, auch ohne Aufschlüsselung jedes einzelnen Sachverhaltes, durch die Anwendung von Wissen, sehr schnell richtige Entscheidungen treffen. Zugleich müssen jedoch auch normative Elemente vom Architekten berücksichtigt werden, die mit großer Wahrscheinlichkeit nicht sofort bei der ersten Neuordnung eines Grundrisses Beachtung finden. Die Gründe hierfür

sind vielfältig. Zum einen ist die Masse der zu berücksichtigenden normativen Elemente (Normen, Gesetze, Verordnungen, Richtlinien) fast unüberschaubar, zum anderen nähert sich der Architekt der Lösung einer Entwurfsaufgabe schrittweise unter Anwendung von Prioritäten. Dabei werden entwickelte Lösungen wiederholt aktualisiert, auf Übereinstimmung zu normativen Elementen überprüft und so dem Entwurfsziel angenähert (vgl. Kap. 2.3.1). Die genannten Schritte der Dekomposition und Neuordnung von Flächen verlaufen während der Entwurfstätigkeit hochgradig parallel und keinesfalls immer strukturiert oder unter expliziter Anwendung von Regeln. Diese scheinbar amorphe Vorgehensweise, die dennoch zu plausiblen Lösungen führt, ist eine der herausragenden Leistungen des menschlichen Gehirns, die durch antrainierte Verhaltensweisen (z.B. das Skizzieren), ohne bewusstes Nachdenken, im Unterbewusstsein ablaufen. Unter „Nachdenken“ sei hier verstanden, dass „... Aussagen unter Nutzung von Schlussfolgerungen zu neuen Gedanken, neuen Aussagen verknüpft werden.“ [Lämmel und Cleve, 2001, S. 171]. Es ist selbstverständlich, dass der Architekt nachdenkt, wenn er Räume neu ordnet, in ihrer Größe und Position verändert oder sie zu neuen Raumeinheiten zusammenfasst.

Möglicherweise ist gerade diese Tatsache der Grund dafür, warum Architekten die Entwurfstätigkeit nicht eindeutig beschreiben können und die Existenz von Entwurfsregeln negieren. Es ist jedoch offensichtlich, dass sich das Entwerfen an solchen Regeln orientiert. Neben den objektiven Regeln, die unabdingbare normative Elemente des Entwurfsprozesses darstellen, sind es vor allem subjektive Regeln, welche die Ausprägung einer Entwurfslösung determinieren. Das Wissen um derartige Regeln wurde im Altertum bereits von Vitruv in eindrucksvoller Weise verkündet [Vitruvius, 1960]. Auch die uralten Entwurfslehren Feng-Shui und Vaastu, die sich heute zunehmender Beliebtheit erfreuen, belegen, dass Architekten nach Regeln entwerfen. Wir wissen, dass altertümliche Tempel und städtische Anlagen der griechischen und römischen Antike nach Regeln aufgebaut worden sind. Auch die Renaissance, mit den herausragenden Bauten von Andrea Palladio beweist die Existenz von Entwurfsregeln. Doch auch heute und in der jüngeren Vergangenheit finden sie sich sowohl in Gebäuden als auch in städtischen Strukturen. Camillo Sitte beschreibt sie 1889 in seiner Schrift „Der Städtebau nach seinen künstlerischen Grundsätzen“ [Sitte, Crasemann Collins et al., 2003]. Viollet-le-Duc und Jean-Nicolas-Louis Durand entwickelten systematische, rationale Entwurfsmethoden, die auf Regeln basieren [Hearn, 2003]. Die quadratischen Proportionen in den Bauten von Oswald Matthias Ungers, die geschwungenen Formen von Zaha Hadid, organische, auf Elementen der Natur beruhende Formen von Santiago Calatrava, das Material bei Hans Kollhoff (Ziegel) oder Shigeru Ban (Papier), sind nichts anderes als die Anwendung verschiedenartiger Entwurfsregeln. Auch Projekte, die unter maßgeblichem Einfluss der Computernutzung entstanden, verwenden Regeln. Kees Christiaanse [Christiaanse, 2004] verwendet Regeln im Städtebau. MVRDV, NOX und Asymptote Architects verwenden Regeln, welche ihre Entwürfe eindeutig charakterisieren. Wie sonst wäre es möglich das Werk eines Architekten zu erkennen, als aus der Extraktion der ihm innewohnenden Merkmale in Form von Regeln? Zwei Vertreter dieser Vorgehensweise, die sich offen zur Existenz von Regeln bekennen, sind Christopher Alexander, welcher Regeln in aller Deutlichkeit beschreibt [Alexander, Ishikawa et al., 1977] und Frank O. Gehry, dessen Gebäude ohne die Verwendung konstruktiver und ästhetischer Regelwerke niemals zu jenen außergewöhnlichen Bauten geworden wären, die sie heute darstellen [Bruggen, 1999; Glymph, Shelden et al., 2002]. Es steht außer Frage, dass Regeln die Architektur in entscheidendem Maße beeinflussen. Seien es

objektive Regeln, deren Anwendung obligatorisch ist, oder auch subjektive Regeln, deren Anwendung vom Individuum abhängt. Für diese Arbeit werden Entwurfsregeln fortan als existent angenommen. Sie determinieren entscheidend die Konzeption eines Entwurfsunterstützenden Systems in der Revitalisierung.



Abb. 3.12 Objektive und subjektive Entwurfsregeln

Die Verwendung objektiver (und subjektiver) Regeln wirft eine weitere Frage auf:

*Welcher Mechanismus sorgt dafür, dass Regeln auf die Flächen, die als veränderliche Parameter den Modellen zu Grunde liegen, angewendet werden können, so dass aus dieser Regelanwendung eine funktionale Lösung entsteht?*

### 3.4.2 Mechanismen zur Problemlösung

Die Anzahl möglicher Flächenneuordnungen innerhalb eines existenten Gebäudes ist bei beiden Modellen in der Regel so groß, dass es kaum möglich erscheint, eine Planungslösung mit herkömmlichen Methoden herbeizuführen, welche objektiv als „die beste“ Lösung deklariert werden kann. Die flächenfüllende Anordnung von nur drei Raumeinheiten innerhalb eines 3 x 5 Einheiten großen Rasters, ergibt theoretisch  $3^{15} = 14.348.907$  Anordnungsmöglichkeiten. Dieses einfache Beispiel verdeutlicht, wie komplex derartige Aufgaben sind, die zudem bereits in einer sehr frühen Phase des Planungsprozesses bearbeitet werden müssen.

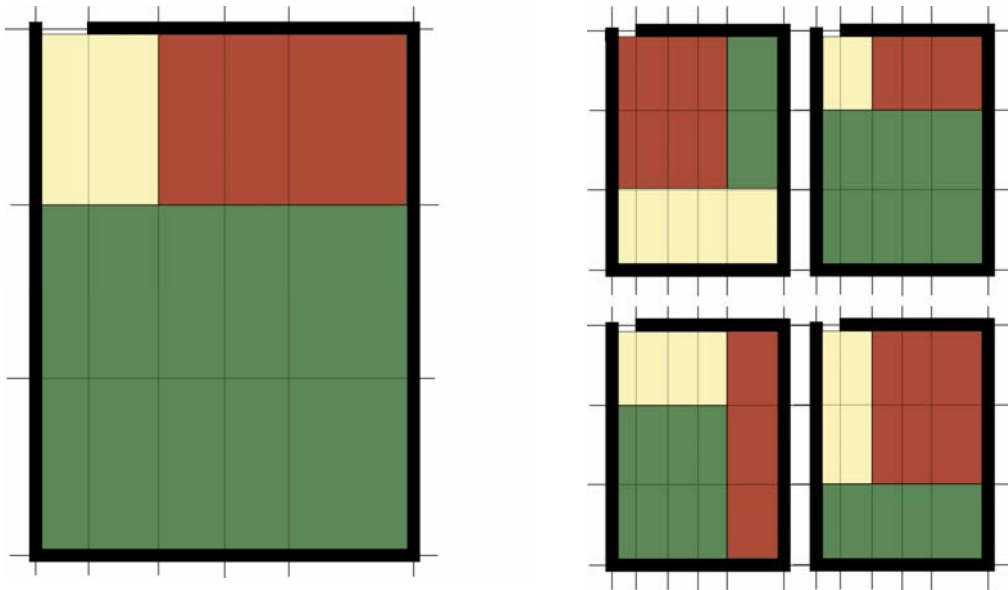


Abb. 3.13 Raster mit 5 x 3 Einheiten, Exemplarische Belegung des Rasters

Es ist unzweifelhaft, dass es einem Architekten trotzdem keine Schwierigkeiten bereitet, eine oder mehrere Lösungen für das gestellte Problem zu finden. Es ist jedoch fraglich, ob die objektiv „beste“ Lösung, welche die Einhaltung aller vom Architekten definierten Bedingungen (Regeln) impliziert, in 14.3 Millionen möglichen Lösungen gefunden wird. Die Anwendung von Regeln, die deklarieren, welche Anforderungen die „beste“ Lösung erfüllen muss, ist Voraussetzung für das Erreichen dieses Zieles. Sicher könnte durch Probieren jede einzelne Lösung erzeugt und anschließend bewertet werden. Diese Vorgehensweise ist jedoch mit hohen Kosten verbunden und daher nicht praktikabel. Des Weiteren kann es auch nicht lösbare Problemkonstellationen geben, für welche annähernde Lösungen gefunden werden müssen. In Entwurfssystemen muss es daher Problemlösungsmechanismen geben, welche durch Verarbeitung von Regeln unter Anwendung von Suchstrategien sämtliche Lösungen und Näherungslösungen erzeugen.

Kapitel 4 beschäftigt sich mit der Frage, wie Wissen, respektive die Regeln zur Findung einer geeigneten Lösung, in der Informationstechnologie repräsentiert werden kann. Das Potential der Informationstechnologie im Bereich der Wissensverarbeitung und der Stand der Forschung zur Entwicklung regelbasierter Entwurfssysteme werden herausgearbeitet. Eine Kurzbeschreibung der Funktionalität heutiger CA(A)D-Systeme veranschaulicht die Diskrepanz zwischen den gebräuchlichen Werkzeugen der Planungspraxis und den Anforderungen, die sich aus der Modellbildung ergeben.



## **4 Informationstechnologie – Potential**

### **4.1 Wissensbasierte Systeme**

In Kapitel 3.4.1 wurde herausgearbeitet, dass Flächen innerhalb eines Gebäudegrundrisses nur dann funktional angeordnet werden können, wenn das Wissen, welche Relationen eine funktionale Lösung beinhaltet, als Gesetzmäßigkeit festgelegt wird. Zur Verwendung von Gesetzmäßigkeiten in einem Computerprogramm müssen diese benannt und formalisiert werden. Kapitel 4.1.1 beschreibt die Grundlagen und Komponenten Wissensbasierter Systeme, mit deren Hilfe Gesetzmäßigkeiten beschrieben und verarbeitet werden können. Aus einem Teilbereich dieser Systeme wird die Systemkonzeption für ZR- und ZA-Modelle abgeleitet.

#### **4.1.1 Grundlagen**

Die Definition dessen, was Wissen bedeutet, ist nicht eindeutig und abschließend geklärt. Häufig wird Wissen mit Intelligenz in Verbindung gebracht und die Einordnung Wissensbasierter Systeme als Teilbereich der Künstlichen Intelligenz (KI) belegt, dass zumindest in der Informatik Zusammenhänge dieser Form hergestellt werden [Waterman, 1986]. Wissensbasierte Systeme (WBS) sind Computerprogramme, die neben Faktenwissen auch mit Unsicherheiten behaftetes heuristisches und subjektives Wissen verarbeiten [Löschner und Menzel, 1993]. Lunze definiert sie als Systeme, durch die Fachleute auf einem speziellen Gebiet bei Entscheidungen unterstützt oder sogar ersetzt werden können [Lunze, 1994]. Nach Waterman zeichnen sich Wissensbasierte Systeme durch explizite Darstellung anwendungsspezifischen Wissens aus [Waterman, 1986]. Als wesentliche Faktoren, die mit dem Einsatz wissensbasierter Techniken verbunden sind, benennt Blumberg unter anderem die Verfügbarkeit und Verfügbarmachung von Wissen, Transformation und Vervielfältigung von Wissen, Sicherung von Wissen und die syste-

matische Bestandsaufnahme von Fachwissen bzw. unternehmensspezifischem Know-How [Thuy und Schnupp, 1989]. Grundlage für die rechnergestützte Wissensverarbeitung ist die Erhebung und anschließende Formalisierung von Expertenwissen, mit dem Ziel der Unterstützung und Ergänzung menschlicher Fachtätigkeit [Herre und Heller, 2001]. Dabei steht nicht ausschließlich die Lösungsfindung für ein bestimmtes Problem im Vordergrund, sondern auch die Reproduzierbarkeit von Entscheidungsprozessen, die in Wissensbasierten Systemen transparent gemacht werden sollen. Die dafür bereitgestellten Komponenten sollen sowohl die Überprüfung von Entscheidungen, als auch die Speicherung des Problemlösungsverhaltens ermöglichen. In der Literatur wird der Prozess der Erhebung und Formalisierung von Wissen in iterierende Phasen aufgeteilt [Rieger, 2001a]:

Phasen [Buchanan, 1993]	Aktion	Iteration	Phasen [Puppe, 1996]	Durchführung	Hilfsmittel
Auswahl:	Bestimme einen geeigneten Problembereich		Problemcharakterisierung	Befragung des Experten durch den Knowledge Engineer	Interviewtechniken und Protokollanalysen
	Auswahl der für die Bearbeitung eines Sachverhaltes relevanten Problemstellung				
Identifikation:	Identifiziere Eigenschaften des Problembereiches				
	Identifikation der Eigenschaften des Problembereiches und Definition der Anforderungen und der Zielstellung.				
Konzeptionalisierung:	Finde Konzept zur Wissensrepräsentation				
	Erarbeitung der für die Problemlösung erforderlichen Konzepte, Relationen, Kontrollmechanismen, Strategien, Teilaufgaben und Nebenbedingungen. Festlegen einer geeigneten Granularität				
Formalisierung:	Konkretisiere die Wissensrepräsentation		Shell-Entwicklung	Auswahl oder Neuentwicklung durch den Knowledge Engineer	Allgemeines Expertensystem-Werkzeug
	Identifikation des prototypischen Problemlösungsprozesses und Konkretisierung der Formalismen zur Wissensrepräsentation. Abbildung der Konzepte und Relationen im gewählten Formalismus.				

Implementierung:	Formuliere Wissen in Regeln		Aufbau der Wissensbasis	Experte ggf. mit Unterstützung durch den Knowledge Engineer	Shell
	Überführung des formalisierten Wissens in ein Computerprogramm.				
Validierung:	Validiere die Regeln und die Wissensrepräsentation		Wartung	Experte ggf. mit Unterstützung durch automatische Analyse-techniken	Falldatenbank
	Überprüfung der Eignung des formalisierten Wissens und Bewertung der Performance des Systems:				

Tab. 4.1 Phasenmodelle der Wissensmodellierung abgeleitet aus [Rieger, 2001a]

### 4.1.2 Komponenten Wissensbasierter Systeme

Wissensbasierte Systeme besitzen einen in der Literatur einheitlich beschriebenen Aufbau, der sich aus fünf Komponenten zusammensetzt:

- Akquisitionskomponente
- Wissensbasis
- Inferenzkomponente
- Dialogkomponente
- Erklärungskomponente

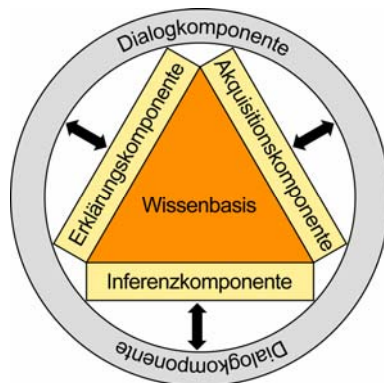


Abb. 4.1 Klassischer Aufbau Wissensbasierter Systeme nach [Waterman, 1986]

#### 4.1.2.1 Erhebung von Wissen - Akquisitionskomponente

Die Aufgabe der Akquisitionskomponente liegt im Aufbau und der Modifizierung der Wissensbasis durch das Hinzufügen, Ändern oder Löschen von Wissensstrukturen [Jung, 2002]. Die Probleme liegen in der Erfassung des Wissens und der notwendigen Formalisierung. Zur Erhebung des Expertenwissens haben sich unterschiedliche Verfahren etabliert. Häufige Verwendung findet das standardisierte Konstruktgitterverfahren, welches in unterschiedlichen Ausprägungen existiert und angewendet wird. Eine weitere Möglichkeit der Wissensakquisition sind Interviewtechniken, zu denen das unstrukturierte Interview, die Introspektion und die Protokollanalyse zählen. Konstruktgitterverfahren und Protokollanalyse können manuell oder automatisiert ausgewertet werden. Dazu stehen Softwarelösungen zur Verfügung, die deren Auswertung unterstützen. Für die Erhebung von Wissen mittels der vorbenannten Verfahren gelten jedoch Charakteristika, die exemplarisch für die Erhebung von Wissen sind. Zu ihnen zählt einerseits der situative Aspekt, d.h. Wissen als Fähigkeit, sich aktiv mit Situationen auseinander zu setzen und zur richtigen Zeit die richtige Vorgehensweise zur Lösung eines Problems beschreiben zu können. Andererseits gehört dazu auch der dynamische Aspekt, nach welchem sich Wissen permanent verändert. Des Weiteren ist Wissen ein höchst persönliches Erhebungs- und Beobachtungsobjekt (subjektiver Aspekt), so dass bei der Akquisition auch den Begleiterscheinungen einer Erhebung (Vertrauen, Suggestion, Relevanz) Beachtung geschenkt werden muss.

Die Erhebung von Wissen kann in drei Grundarten unterschieden werden: direkte, indirekte und automatische Wissenserhebung.

Grundarten der Wissenserhebung	Ausprägung	Problem in der Architektur
direkt	Der Experte formalisiert das Wissen selbst. <u>Problem:</u> Vorstrukturierte Eingabesysteme sind notwendig. Komplexität des Wissensbereiches muss überschaubar sein. <u>Durchführung:</u> Softwaresysteme	Der Experte (Architekt), weiß nicht, wie das Wissen formuliert werden muss, so dass es in Wissensbasierten Systemen verwendet werden kann.
indirekt	Ein Wissensingenieur befragt einen Experten und formalisiert die Ergebnisse für das Wissensbasierte System <u>Problem:</u> Methodisch aufwendig und fehleranfällig. <u>Durchführung:</u> Interviewtechniken	Der Wissensingenieur kennt nicht die richtigen Fragen, da er mit dem Planungsprozess nicht vertraut ist. Diskrepanz zwischen Theorie und Praxis.
automatisch	Das System extrahiert sich das notwendige Wissen selbstständig aus Falldaten oder Literatur <u>Problem:</u> Erfordert einen überschaubaren Wissensbereich. <u>Durchführung:</u> Forschungsansätze, keine ausgereiften Systeme vorhanden.	Eine theoretisch gut geeignete Form der Wissenserhebung, die jedoch deshalb nicht praktikabel ist, da es keine allgemeingültigen Softwaresysteme gibt, die derartiges Wissen aus Falldaten extrahieren könnten.

Tab. 4.2 Formen der Wissenserhebung in Anlehnung an [Herre und Heller, 2001]

#### 4.1.2.2 Speicherung von Wissen - Wissensbasis

Die Wissensbasis ist jene Komponente, die das Wissen zur Lösung einer Problemstellung enthält. Die Struktur der enthaltenen Daten ist unterschiedlich. Häufig ist Wissen in Form von Produktionsregeln und Fakten (Maschinenbau, Fahrzeugtechnik) formalisiert, deren Implementierung durch geeignete Programmiersprachen (z.B. Prolog) unterstützt wird [Jung, 2002]. Für den Aufbau Wissensbasierter Systeme ist es von Bedeutung, die Struktur des zu verarbeitenden Wissens zu definieren. Sie entscheidet über die Form der Repräsentation des Wissens im Zielsystem [Jung, 2002]. Rieger charakterisiert die Repräsentation von Wissen folgendermaßen:

Arten von Wissen:	Merkmale
Wissen über Objekte	Attributierung von Wissens-elementen einschließlich Wertvergabe („Fläche ist 20 m <sup>2</sup> groß“).
Wissen über Zusammenhänge	taxonomisch („ist vom Typ HNF“) oder kausal, als Zusammenhang zwischen Ursache und Wirkung („HNF ermittelt sich aus NF und VF“, „wenn VF zu groß, dann Lösung unwirtschaftlich“).
Wissen über Abläufe	Schritte und Reihenfolge einer Problemlösung („wenn VF zu groß, prüfe Abstand der Erschließungskerne, dann Breite der VF, ...“).
Wissen über die Anwendung von Ablaufwissen	Anwendung von Metawissen, d.h. Kenntnisse darüber, wann welches Wissen wie angewendet wird („wenn Treppe geplant wird, dann §31 MBO, DIN 18025, 18064, 18065 beachten“)

Tab. 4.3 Wissensarten in Anlehnung an [Rieger, 2001b]

#### Repräsentation von Wissen:

Wissen kann in unterschiedlicher Form repräsentiert werden. Die wesentlichen Repräsentationsformen sind (Aussagen-/Prädikaten)-Logik, semantische Netze, AW-Tupel und OAW-Tripel, Frames und Objekte, Regeln und Constraints.

Die Logik zeichnet sich durch eine Menge von Fakten oder Aussagen aus, die wahr oder falsch sein können. Es können allgemeingültige Ableitungsregeln (modus ponens, Resolution, Horn-Klauseln) angewandt werden, mit denen neue Fakten aus alten Fakten gefolgert werden können.

Semantische Netze sind eine modellhafte Anlehnung an das menschliche assoziative Gedächtnis. Sie sind gerichtete Graphen aus Knoten und Kanten, in denen die Knoten Objekte (Individuen) oder Konzepte (Klassen) und Ereignisse (events) repräsentieren und die Kanten die Beziehung zwischen den Knoten definieren. Kanten semantischer Netze können generalisierende (IS-A), erklärende (HAS-A), zerlegende (HAS-PART) oder frei zu definierende (z.B.: DIENT-ZUM) Zuordnungen darstellen.

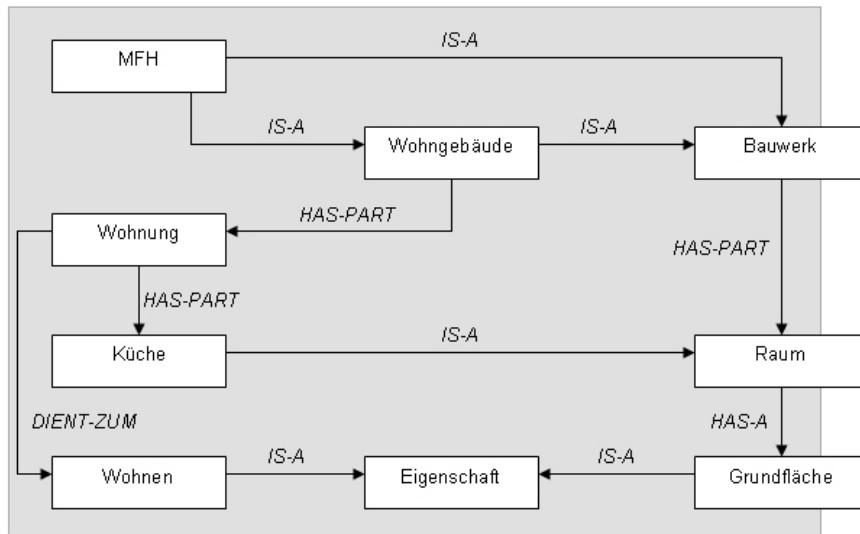


Abb. 4.2 Semantisches Netz eines Mehrfamilienhauses (MFH)

AW-Tupel und OAW-Tripel sind Spezialfälle semantischer Netze, die häufig regelbasierten Systemen zu Grunde liegen. Die Knoten können in Form von Objekten, Attributen oder Werten auftreten, wohingegen die Kanten wiederum die Beziehungen zwischen den Knoten verdeutlichen.

Frames sind Datenstrukturen zur Repräsentation stereotypischer Situationen, die zugehörige Informationen deklarativer und prozeduraler Art in Fächern (Slots) sammeln [Rieger, 2001b]. Frames können hierarchisch angeordnet werden und Klassen, Unterklassen oder konkrete Objekte sein. Ein allgemeineres Konzept bilden die Objekte, deren Charakteristika Datenkapselung, Vererbung und Polymorphismus sind.

Regeln eignen sich für die Repräsentation prozeduralen Wissens. Sie bestehen aus einer Prämisse (Wenn-Teil), die erfüllt sein muss, bevor die Konklusion (Dann-Teil) abgeleitet werden kann. Dabei kann die Konklusion gleichzeitig eine Komponente der Prämisse einer weiteren Regel sein. Die Prämisse bezieht sich auf den Wahrheitsgehalt von Fakten (Aussagen, AW-Tupel, OAW-Tripel, Slots, Objekt-Instanzvariablen) und kann sowohl logische Verknüpfungen (**AND**, **OR**, **NOT**) als auch Variablen beinhalten. Aus der Prämisse folgt normalerweise nur eine einzige Konklusion, die eventuell mehrere Aktionen auslöst. Regeln in Form von Constraints (Nebenbedingungen) sind wesentlicher Bestandteil der erarbeiteten Systemkonzeption (vgl. Kap. 5).

Beispiel:

WENN (Prämisse) - DANN (Konklusion)

WENN Baugebiet = allgemeines Wohngebiet (WA) - DANN Geschossflächenzahl = 1,2

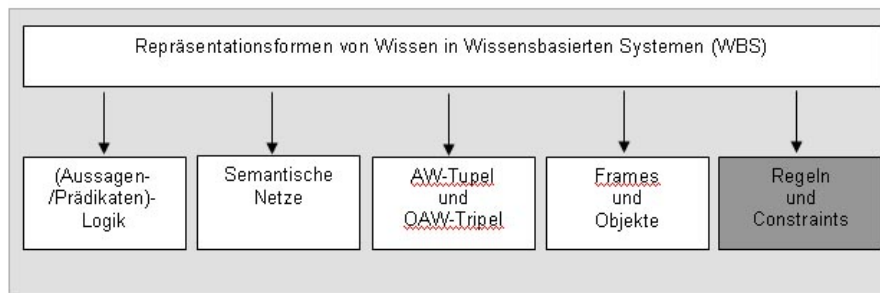


Abb. 4.3 Repräsentation von Wissen in WBS

#### 4.1.2.3 Verarbeitung von Wissen - Inferenzkomponente

Die Verarbeitung des gespeicherten Wissens der Wissensbasis erfolgt durch die Inferenzkomponente. Sie interpretiert das Expertenwissen zur Lösung des vom Benutzer spezifizierten Problems unter Verwendung bestimmter Strategien (Inferenzmechanismen). Die erzeugte Problemlösung wird an die Erklärungskomponente weitergeleitet. Die während des Problemlösungsprozesses erzeugten Fakten werden temporär oder auch dauerhaft in der Wissensbasis abgelegt. Auf die durch Regeln vorgegebenen Fakten (Prämissen) der Wissensbasis werden unterschiedliche Mechanismen angewendet, die diese Fakten auswerten und zu einer Konklusion führen. Man unterscheidet zwischen Vorwärts- und Rückwärts-Verkettung als grundsätzliche Suchrichtungen. Die Inferenzkomponente beginnt mit der Suche nach Regeln, welche die Zielvorgabe im Konklusionsteil (Dann-Teil) der Regeln beinhalten. Wenn Regeln gefunden worden sind, wird deren Anwendbarkeit untersucht, d.h. es wird überprüft, ob der Wenn-Teil der Regel erfüllt ist. Bei Erfüllung des Wenn-Teils der Regel wird diese angewandt, bei Nicht-Erfüllung wird der Wenn-Teil der Regel zu einem Unterziel erklärt und die Inferenzkomponente sucht nach Regeln, die diesen Wenn-Teil erfüllen können. Ist keine Regel mehr zu finden, fragt die Inferenzkomponente den Benutzer nach Werteingaben. Diese Werteingaben und die Ergebnisse ausgeführter Regeln werden der Wissensbasis zugeführt. Die als Folge der Zuführung anwendbaren Regeln werden ausgeführt und ihre Ergebnisse wiederum in die Wissensbasis eingefügt. Der Prozess ist beendet, sobald das Ziel erfüllt ist, keine Regel mehr angewandt oder der Benutzer keine Werteingabe mehr durchführen kann. Sowohl bei der Vorwärts- als auch bei der Rückwärts-Verkettung kann es passieren, dass gleichzeitig mehrere Regeln oder Fakten für die Lösungssuche in Frage kommen. In diesem Fall muss eine Entscheidung herbeigeführt werden, welche Regel oder welches Faktum zuerst untersucht werden soll. In Wissensbasierten Systemen werden zur Lösung dieser Konfliktsituationen Suchstrategien (z.B. Tiefensuche, Breitensuche, Bestensuche) eingesetzt. Zusammenfassend lässt sich festhalten, dass die Inferenzkomponente die Aufgabe hat, Wertbelegungen von Fakten zu überprüfen und zu schlussfolgern, welche neuen Fakten aus bekannten Fakten entstehen. Ferner sucht die Inferenzkomponente nach der Beantwortung von Fragen, indem sie geeignete Regeln überprüft und diese gegebenenfalls miteinander verknüpft. Zur Verkettung der Regeln benutzt sie die Vorwärts- oder Rückwärts-Verkettung unter Einbeziehung unterschiedlicher Suchstrategien.

#### **4.1.2.4 Interaktion mit dem Benutzer - Dialogkomponente**

Die Dialogkomponente führt den Dialog mit dem Benutzer des Systems. Ihre Aufgabe besteht in der verständlichen Aufarbeitung der systeminternen Darstellung des Wissens für den Benutzer. Zusätzlich zur Aufarbeitung der Ergebnisse des Lösungsprozesses ist die Dialogkomponente für Anfragen des Benutzers an das System zuständig.

#### **4.1.2.5 Darstellung der Ergebnisse - Erklärungskomponente**

Die Erklärungskomponente macht die Vorgehensweise des Systems transparent, indem sie den Benutzer über Folgerungsmechanismen informiert, die zu einem bestimmten Ergebnis geführt haben. Durch spezielle Protokollfunktionen ist die Erklärungskomponente in der Lage, Fragestellungen und Ergebnisse zu begründen.

#### **Kurzzusammenfassung**

Der entscheidende Unterschied zwischen konventionellen und Wissensbasierten Systemen liegt in der Aufspaltung des Algorithmus des Wissensbasierten Systems in eine Komponente, die problemspezifisches Wissen (z.B. in Form von Regeln und Fakten) repräsentiert, und in eine von dem Problem unabhängige Inferenzkomponente zur Problemlösung. Während in konventionellen Systemen die Regeln zur Lösung eines Problems fest verankert sind, ist die Wissensbasis des Wissensbasierten Systems variabel. Mit Hilfe der Inferenzkomponente wird ein „temporäres“ Programm zur Lösung der Aufgabenstellung aus den in der Wissensbasis zur Verfügung stehenden Regeln erstellt. Dabei kann das Wissen zur Lösung eines Problems in Form von vielen voneinander unabhängigen Einzelschritten repräsentiert werden. Diese Einzelschritte sind nicht fest miteinander verknüpft, so dass alle kombinatorisch möglichen Fragestellungen vom System abgedeckt werden können. Die Regelbasen können derart aufgebaut sein, dass die Erweiterbarkeit und Anpassung durch den Nutzer gegeben ist. Regeln sind editierbar, da keine Kompilierung erforderlich ist. Die Nachteile gegenüber konventionellen Systemen liegen in der höheren Laufzeit der Programme, da die Durchsuchung der Regelbasis sowie die Anwendung der Regeln zeitaufwändig ist. Ferner sind, zur Gewährleistung breiter Anwendbarkeit, die Verwendung findenden Suchverfahren so strukturiert, dass häufig nur einfache Problemstellungen bearbeitet werden können. Wissensbasierte Ansätze gelangten in der Vergangenheit häufig in Expertensystemen zum Einsatz. Bei Leppich [Leppich, 1999] und Schumann [Schumann, 2001] findet sich ein Überblick zu dieser Thematik.



## 4.2 Wissensbasierte Systeme in der Architektur

### 4.2.1 Planungspraxis

Auf Grund der in Kapitel 2.3 beschriebenen diversifizierenden Methodik des Planens und Entwerfens ist es nicht verwunderlich, dass es weder in der Neubauplanung noch in der Revitalisierung Planungsunterstützende Entwurfssysteme gibt, die in der Praxis uneingeschränkt eingesetzt werden könnten. Das digitale Handwerkszeug des Architekten für den Gebäudeentwurf beschränkt sich nach wie vor auf die Verwendung von CAAD-Systemen, die jedoch nur wenig inhaltliche Unterstützung bieten. Diese Systeme sind in ihrer grundsätzlichen Funktionalität gleich. Es wird hier nicht der Versuch unternommen, eine Kategorisierung der CAAD-Systeme zu entwickeln. Es ist jedoch erwähnenswert, dass der weitaus größte Teil der professionellen CAAD-Systeme sich nicht am Prozess der Planung, sondern am Prozess der Umsetzung der Planung, d.h. des Zeichnens und Konstruierens bereits vorhandener Planungsergebnisse, orientiert. CA(A)D-Systeme, die nicht für Architekten, sondern für Bauherren entwickelt werden, orientieren sich interessanterweise eher an der dem Architekten zugeschriebenen Planungstätigkeit als an der zeichnerischen Umsetzung einer bereits entwickelten Lösung (z.B.: 3D Home Architect, 3D Home Designer, FloorPlan 3D, u.a.). Über den Grund, warum CAAD-Systeme in der Architektur diese inhaltliche Unterstützung nicht bieten, kann nur gemutmaßt werden, da Forschungsergebnisse hierzu nicht vorliegen. Die wenigen für die Praxis entwickelten Systeme (z.B. acadGraph Alberti [Insider, 1998]) auf der Grundlage von [Steinmann, 1997], scheiterten - neben ihrem unzulänglichen technischen Potential - sicher auch am begrenzten Willen der Architekten diese Systeme tatsächlich zu erproben. Das Architekturstudenten, die wenig Erfahrung mit professionellen CAAD-Systemen sammeln konnten, ein System zur Unterstützung der Planung völlig anders konzipieren würden, belegen die Ergebnisse „Semsterbegleitender Entwürfe“ an der Bauhaus-Universität Weimar [Donath und Lömker, 2003a].

Die inhaltliche Unterstützung der Planung in CAD-Systemen, die in fachfremden Disziplinen (Maschinenbau, Fahrzeugtechnik, Luft- und Raumfahrttechnik) benutzt werden könnte jedoch auch in der Architektur zunehmend an Bedeutung gewinnen. Systeme wie CATIA (Hersteller Dassault) oder Pro/ENGINEER (Hersteller PTC), die ebenso wie CAAD-Systeme mit parametrischer Geometrie arbeiten, bieten zudem die Option, Geometrie mit Optimierungsalgorithmen zu verknüpfen. Eine geeignete Formalisierung vorausgesetzt, bietet diese Vorgehensweise durchaus Ansätze für die Integration von Wissen in Entwurfsunterstützende Systeme (vgl. Kap. 7.1). Dies ist deshalb von Belang, da diese Systeme zunehmend in der Architektur Anwendung finden. Die CAAD-Software „Digital Project“ von Gehry Technologies integriert beispielsweise CATIA's „Knowledge Template“ und „Knowledge Advisor“, so dass damit auch in der Architektur ein Wissensbasiertes System zur Verfügung steht [Gehry, 2004].

## 4.2.2 Stand der Forschung und eigene Vorarbeiten

Forschung zu Wissensbasierten Systemen in der Architektur gibt es seit mehr als vier Jahrzehnten. Spürbaren Einfluss auf die Entwicklung der CAAD-Systeme hat diese Forschung jedoch nicht genommen. In Deutschland wurde insbesondere in den 90er-Jahren umfangreiche Forschung betrieben, so z.B. im Verbundprojekt Fabel [Gauchel, 1990; Lenart, 1991]. Da die Forschung in der Regel auf den konkreten Einzelfall bezogene Resultate lieferte, würde eine Auflistung der Ergebnisse zu keinem Erkenntnisgewinn für die in dieser Arbeit behandelte Problematik führen. Es sei daher auf die umfangreichen Publikation verwiesen, die unter anderem bei Tasli [Tasli, 1997] und Michalek [Michalek, 2001] vermerkt sind. Im außereuropäischen Ausland sind hier insbesondere die Arbeiten von Gianfranco Carrara, Chuck Eastman, John Gero und Yehuda Kalay von Bedeutung, die seit mehreren Jahrzehnten in diesem Bereich tätig sind und die Theoriebildung maßgeblich beeinflussten. Entwurfssysteme mit wissensbasierten Komponenten für die Revitalisierung gibt es bislang nicht. Die Entwicklung der in Kapitel 5 vorgestellten Systemkonzeption orientiert sich an den Ergebnissen der Forschung zu Wissensbasierten Systemen in der Architektur sowie an den in Kapitel 2 und 3 erarbeiteten Anforderungen des Planungsprozesses und der Revitalisierungsstrategien. Zur prototypischen Umsetzung der Systemkonzeption wird unter anderem eine Modellierungssprache verwendet, die in der architektonischen Fachliteratur bislang nicht erwähnt wurde. Die Besonderheiten dieser Modellierungssprache werden in Kapitel 7.3 erläutert.

Als eigene Vorarbeiten in diesem Bereich wurden zwei Module für die Anwendung der Abstandsflächenverordnung (SPACEplan, BauNVO) konzipiert und programmtechnisch umgesetzt [Lömker, 2003]. Ein weiteres Modul wurde mit Hilfe der „Knowledge Advisor-Komponente“ von CATIA entwickelt.

## 4.3 Bewertung aus architektonischer Sicht – Bescheidenheit

Obwohl die Grundlage der Wissensbasierten Systeme, das Wissen, nicht eindeutig definiert ist, wird in der Literatur ein einheitlicher, klarer und funktionaler Aufbau Wissensbasierter Systeme beschrieben. Es gibt Komponenten zur Abbildung (Akquisition), Speicherung (Basis), Verarbeitung (Inferenz) und verständlichen Aufarbeitung (Dialog) von Wissensstrukturen und zur Begründung (Erklärung), der aus konkreten Problemen resultierenden Ergebnisse. Die Praxis zeigt, dass sich dieser idealtypische Aufbau jedoch selbst in den Haupteinsatzbereichen (Maschinenbau, Fahrzeugbau, Medizin) nicht verallgemeinert anwenden lässt. So unterliegt der Einsatz Wissensbasierter Systeme immer der Voraussetzung, dass der Problembereich eindeutig definiert und formalisiert werden kann. Ferner kann ein solches System nur durch die Zusammenarbeit von Experten aus dem Bereich des Knowledge-Engineering mit Experten aus der jeweiligen Domäne realisiert werden (Puppe nach [Rieger, 2001a]) und [ILOG, 2005]. Es ist des Weiteren festzuhalten, dass viele alltägliche Probleme der Planungspraxis in der Architektur derart komplex sind, dass ihre Umsetzung und Lösung mit gängigen Methoden Wissensbasierter Systeme zu aufwändig oder auch gar nicht möglich ist [Gritzmann und Brandenburg, 2003].

Die Beschreibung der Systeme erweckt den Eindruck, dass diese auf Grund ihrer klaren Struktur, allgemein verfügbar und vom Architekten unmittelbar einzusetzen wären. Dies ist mitnichten der Fall. In der architektonischen Planungspraxis spielen Wissensbasierte Systeme auch deshalb keine Rolle. Die Ursachen hierfür liegen im hohen Anspruch der aus der Benutzung dieser Programme resultiert, sowie der zur Implementierung von Wissen notwendigen, jedoch fehlenden methodischen Vorgehensweise des Architekten im Planungsprozess (vgl. Kap. 2.3). Ein Wissensbasiertes System kann nur dann unterstützend im Planungsprozess eingesetzt werden, wenn die Regeln dieses Prozesses transparent gemacht werden und die Systeme vom Architekten selbstständig bedient und gepflegt werden können. Da viele Architekten diese Transparenz scheuen, kann die Integration der Systeme in den Planungsprozess im Grunde genommen nur nach subjektiven Kriterien erfolgen. Der in Kapitel 4.1 erläuterte exemplarische Aufbau eines Wissensbasierten Systems hat dennoch Bedeutung für die konzeptionelle Verwendung dieser Technologie in der Revitalisierung.

## 4.4 Zusammenfassung und Schlussfolgerung

Computerprogramme, welche den Architekten planerisch bei der Umstrukturierung, dem Erhalt und der Weiternutzung von Bausubstanz unterstützen, erfordern die Integration formalisierten Wissens. Das Wissen, wie zu einem spezifischen Problem eine Lösung gefunden werden kann, basiert auf Erfahrung, Erlerntem und auch auf der Verwendung von Analogien aus anderen Disziplinen. Es gibt Wissensarten, die sich mit Hilfe von Regeln erfassen lassen. Diese Regeln werden alltäglich vom Architekten bei der Lösung von Entwurfsproblemen verwendet. Planungs- oder Entwurfssysteme, wie sie in der Architektur gebräuchlich sind, kennen üblicherweise keine Regeln.

### Kontext

Die Planung ist ein übergeordneter Prozess, innerhalb dessen das Entwerfen (von Neubau- und Revitalisierungsmaßnahmen) als ergebnisorientierter, kontinuierlich zu aktualisierender Prozess definiert werden kann (vgl. Kap. 2.3). Infolgedessen muss ein System zur Entwurfsunterstützung in der Revitalisierung als untergeordneter Bestandteil (Modul) eines Planungssystems konzipiert werden.

Planungssystem	Ein Planungssystem ist ein digitales Hilfsmittel, dass den Architekten kontextbezogen bei der Bearbeitung von Planungsleistungen, durch die Bereitstellung von Informationen und die Erzeugung von Vorschlägen zur Lösung einer Planungsaufgabe, unterstützt. Der Begriff „Planungssystem“ bezieht sich auf die ganzheitliche am Planungsprozess orientierte Unterstützung durch Computerprogramme bei der Bearbeitung von Planungsleistungen, die ein Architekt zu erbringen hat.
----------------	--

Planungsprozess	Der Planungsprozess beinhaltet die Gesamtheit der Planungsleistungen, die ein Architekt zu erbringen hat. Der Planungsprozess schließt auch die Ergebnisse der Leistungen Dritter ein, die zur Erlangung von Planungssicherheit relevant sind.
Planungssicherheit	Planungssicherheit bedeutet, dass ein Planungssystem dem Architekten bei der Entscheidungsfindung während der Bearbeitung der Planungsleistungen Unterstützung bietet, so dass ein derartig entwickelter Lösungsweg als plausibel, begründbar und nachvollziehbar angesehen werden kann.
Plausibilität	Ein entwickelter Lösungsweg ist dann als plausibel anzusehen, wenn die erzeugte Lösung den für die erfolgreiche Lösung einer Bauaufgabe zu beachtenden Regeln entspricht und die Lösungsfindung in geeigneter Form nachvollziehbar und begründbar ist.

Das Neuordnen der Flächen bei ZR- und ZA-Modellen allein erzeugt keine plausible Planungslösung. Es ist unzweifelhaft, dass viele weitere Faktoren, die bei den Modellen keine Berücksichtigung finden, Einfluss auf die Entscheidung ausüben, ob und wie ein Gebäude revitalisiert wird. Die Erstellung von Werkzeugen, die den Planer bei der Ermittlung von Lösungswegen zu einer Fachfrage inhaltlich unterstützen und die Nachvollziehbarkeit und Richtigkeit einer Planungsentscheidung gewährleisten und plausibel darlegen, orientiert sich daher an den nachfolgend genannten Thesen [Donath und Lömker, 2003b]:

Planungsprozess	<p>Dem Planer fehlen digitale Werkzeuge zur Verbesserung seiner Argumentationsfähigkeit, zur Nachvollziehbarkeit einer Planungslösung und zur fachlich detaillierten Begründung der Planungsabsicht.</p> <p>Es gibt zahlreiche Probleme in der Planung und baulichen Umsetzung, die heute keinerlei digitale Unterstützung erfahren.</p> <p>Viele dieser Probleme sind analysierbar, abgrenzbar und durch Programmsysteme unterstützbar.</p> <p>Es gibt wiederkehrende Prozesse, die bei jeder Planungsaufgabe auftreten und digital unterstützt werden können. Diese Prozesse resultieren aus den Erfordernissen der Planungspraxis.</p>
Modellbildung	<p>Existente Modellverwaltungssysteme, Prozessbeschreibungsmodelle, Datenbanken und Systeme des verteilten Arbeitens bieten eine ausreichende Basis für die Schaffung Planungsunterstützender Werkzeuge.</p> <p>Diese Planungsunterstützenden Werkzeuge werden als überschaubare, singuläre und in sich thematisch abgeschlossene Einheiten (Module) konzipiert.</p>

Akzeptanz	<p>Es gibt die Diskrepanz zwischen programmierter Leistungsfähigkeit existenter Systeme und nachweisbarer Nichtauslastung/-benutzung durch die Anwender.</p> <p>Der Planer benötigt getrennt nutzbare, in sich abgeschlossene Software, die ihn bei der Lösung einer Fachfrage unterstützt.</p> <p>Diese Software muss einfach zu benutzen sein und sich an der Vorgehensweise des Planers und am Planungsprozess orientieren.</p>
-----------	--

Die nachfolgende Beschreibung listet die Komponenten Wissensbasierter Systeme und ihre Entsprechung in der Konzeption für die wissensbasierte Unterstützung der ZR- und ZA-Modelle:

#### **Erhebung von Wissen (Eingabemodell)**

Im Entwurfsprozess werden von Architekten unterschiedliche Methoden und Techniken eingesetzt, die nicht verallgemeinert werden können (vgl. Kap. 2.3). Ein Entwurfsunterstützendes System für die Revitalisierung muss daher die Flexibilität beinhalten, neben objektivem Wissen auch subjektives und dynamisches Wissen zu speichern und zu verarbeiten. Die Erhebung des Wissens, welches für die Lösung einer Entwurfsaufgabe notwendig ist, stellt dabei die größte Herausforderung dar. Als Parameter für das Eingabemodell werden Nachbarschaftsbeziehungen zwischen Flächen [Eastman, 1973] und Design Constraints verwendet. Diese Parameter beruhen auf der Gegebenheit, dass das Wissen um eine gute Lösung in jedem funktionalen Grundriss bereits enthalten ist. Der Bestand kann daher als Instrument genutzt werden, um Beziehungen zwischen Flächen zu definieren und diese Beziehungen in Regeln zu überführen.

#### **Speicherung von Wissen (Optimierungsmodell)**

Für die Speicherung und Repräsentation von Wissen werden Regeln in Form von Constraints (Nebenbedingungen) verwendet. Diese werden „allgemeingültig“ mathematisch formalisiert, so dass eine prototypische Verwendung unabhängig von einer bestimmten Sprache oder Entwicklungsumgebung erfolgen kann. Die Regeln und Constraints repräsentieren Wissen über Objekte sowie Wissen über Zusammenhänge zwischen Objekten.

#### **Verarbeitung von Wissen (Optimierungsmodell)**

Den Kernpunkt des Wissensbasierten Systems bildet die Inferenzkomponente, welche in der Systemkonzeption durch die Verwendung von Optimierungswerkzeugen gebildet wird. Diese werden in Kapitel 5 eingehend erläutert.

#### **Interaktion mit dem Benutzer (Geometrie- und Optimierungsmodell)**

Die Definition der Regeln und Constraints erfolgt in Textform. Die Darstellung der Optimierungsergebnisse wird numerisch oder graphisch realisiert. Für die Integration eines Wissensbasierten Systems in ein bestehendes CAAD-System ist die regelbasierte parametrisch-assoziative

Modellierung von besonderer Bedeutung [Lömker, 2003]. Dieses Konzept wird im CATIA-Prototyp (vgl. Kap. 7.1) evaluiert.

- Parametrisierung** Eine Konstruktion gilt dann als parametrisch, wenn die Dimensionierung eines Objektes zumindest teilweise über benannte Maßangaben erfolgen kann und die Zusammenhänge der Gesamtkonstruktion bei Veränderungen einzelner Parameter erhalten bleiben. Ein auf parametrischer Modellierung basierendes Objekt kann auf diese Weise zu jedem Zeitpunkt verändert und an die Erfordernisse der Planung angepasst werden. Die in CAAD-Systemen der heutigen Generation Verwendung findenden parametrischen Konstruktionen bestehen im Regelfall aus Standard-Objekten wie z.B. Wänden, Fenstern, Türen und Treppen. Die Konstruktionsbeziehungen der Einzelkomponenten sind üblicherweise fest verankert und können weder verändert noch auf andere Objekte übertragen werden. Dieser Ansatz besitzt großes Potenzial, welches auch auf andere Objekte (z.B. Flächen) übertragen werden kann. Voraussetzung dafür ist die Möglichkeit, alle Objekte eines CAAD-Modells grundsätzlich parametrisch definieren und bearbeiten zu können, sowie das Vorhandensein von Assoziativität zwischen diesen Objekten.
- Assoziativität** Die Kombination des parametrischen Ansatzes mit dem assoziativen Aufbau eines Objektes ermöglicht die Erstellung von Konstruktionen, welche sich intelligent verhalten und dem Planer ein Höchstmaß an Überprüfungsarbeit abnehmen [Glymph, Shelden et al., 2002]. Durch die Assoziativität zwischen zwei Objekten wirken sich parametrische Veränderungen eines Bauteiles auf die Parameter eines anderen Objektes aus.
- Regeln** Bei der Betrachtung assoziativ-parametrischer Strukturen wird deutlich, dass eine funktionsfähige Konstruktion nur durch die Erzeugung von Beziehungen zwischen mehreren Objekten entstehen kann. Diese Beziehungen können in Form von Regeln mit Hilfe einer wissensbasierten Komponente erstellt werden. Diese gewährleistet die Erfassung und Interpretation von Regeln sowie die kontinuierliche Evaluierung der Regel im Gesamtkontext. Die während des gesamten Planungsprozesses durchgeführte Evaluierung der Konstruktion führt durch die Anwendung entsprechender Prüfmechanismen zur Einhaltung festgelegter Regeln. Da nicht auszuschließen ist, dass konkurrierende Regeln existieren (vgl. Kap. 5.3.1), kann durch die Verwendung gewichteter Zielfunktionen auf deren Erfüllungsgrad und -priorität eingewirkt werden.

### **Darstellung der Ergebnisse (Geometrie- und Performancemodell)**

Die Darstellung der Ergebnisse erfolgt in Abhängigkeit von den definierten Zielfunktionen. Bei Optimierungsproblemen mit einer oder mehreren zu minimierenden oder zu maximierenden Zielfunktionen (Multi-Objective Optimization), kann die Performance textlich oder graphisch abgebildet werden. Bei letzteren empfiehlt sich die Bildung eines übergeordneten Zieles als gewichtete Summe der Teilziele (vgl. Kap. 5.3). Deren Erfüllungsgrade können somit einzeln abgebildet werden und gewährleisten die detaillierte Beurteilung des Erreichungsgrades jedes einzelnen Teilzieles sowie des Gesamtzieles. Bei Constraint Satisfaction Problemen, d.h. bei Optimierungsproblemen ohne Zielfunktion, ist die Performance irrelevant, da derartige Probleme immer die vollständige Erfüllung aller Constraints voraussetzen.

## 5 Systemkonzeption

Die Zusammenhänge zwischen den Räumen eines Gebäudes sind vielschichtig und im Einzelfall nur schwer zu definieren. Es ist offensichtlich, dass das Entwerfen einerseits an Regeln gebunden ist und andererseits an etwas, das möglicherweise mit den Begriffen Intuition, Kreativität und auch Erfahrung bezeichnet werden kann [Lömker, 2005]. Der Architekt tut in vielen Fällen das Richtige, ohne darüber nachdenken zu müssen, warum er es tut. Es gibt eine Vielzahl von Untersuchungen, insbesondere aus den 70er Jahren, in denen versucht wurde, das Phänomen des Entwerfens zu erklären und wissenschaftliche Theorien daraus abzuleiten. Allen Untersuchungen ist gemein, dass aus ihnen keine generelle Vorgehensweise für das Entwerfen abgeleitet werden kann. Es ist außerdem nicht gelungen, Vorgänge während des Entwerfens in Gänze wissenschaftlich herzuleiten oder zu definieren. Gegenstand der Untersuchungen dieser Arbeit soll daher nicht der erneute Versuch sein, dies zu tun. Dennoch ist es für die Erarbeitung der Systemkonzeption von Bedeutung, Vorgehensmerkmale aus dem Entwurfsprozess zu extrahieren, die offensichtlich von Architekten bei der Organisation von Grundrissen eingesetzt werden. Diese Merkmale, die für die Entwicklung eines Entwurfssystems für die Revitalisierung unabdingbar sind, können aus existenten Grundrissen abgeleitet werden. Wesentlich ist, dass mit Hilfe der ZR- und ZA-Modelle architektonische Zusammenhänge, gleich welcher Art, beschrieben und in Optimierungsprozessen als Parameter eingesetzt werden können. Vereinfacht werden hier Beziehungen zwischen Raumeinheiten verwendet, die offensichtlich Relevanz bei der Organisation von Grundrissen besitzen.

Die Systemkonzeption beschreibt die konzeptionelle Übersetzung der ZR- und ZA-Modelle in Optimierungsmodelle, sowie den Aufbau der Eingabe-, Geometrie- und Performancemodelle.



## 5.1 Eingabemodell

Die Grundlage der Optimierungsmodelle sind Daten über die Eigenschaften der einzelnen Flächen innerhalb des Gesamtmodells. Zu diesen Eigenschaften gehören Aussagen über die Lage einer Fläche, deren Größe oder die Benachbarung zu anderen Flächen. Da Raumprogramme in der Regel Daten über Beziehungen, d.h. die Lage und Nachbarschaftsverhältnisse von Flächen, nicht enthalten, ist die Zuordnung dieser Eigenschaften Aufgabe des Architekten und wesentlicher Bestandteil des Entwurfprozesses. Es ist jedoch offensichtlich, dass eine Reihe von Eigenschaften unveränderlich oder zumindest häufig in bereits gebauten Entwürfen zu finden ist [Stamm-Teske und Donath, 2006]. Eine Evaluation bestehender Gebäude zur Überprüfung der darin enthaltenen Raumeigenschaften bildet demnach die Grundlage der Aufstellung von Regeln für neu zu definierende Raumprogramme.

### 5.1.1 Beziehungen



Das Zerstörungsreiche Modell benutzt Daten über Flächen, die dem Raumprogramm entnommen werden, welches herkömmlich zu Beginn einer Revitalisierungsmaßnahme vorliegt (vgl. Kap. 2.2.2). Das Raumprogramm listet die notwendigen Räume und die damit verbundene erforderliche Nutzfläche auf. Die Aufgabe des Architekten ist es, diese Flächen zu ordnen. Da verbindliche Aussagen über die Beziehungen zwischen einzelnen Flächen nicht vorliegen, ergänzt der Architekt diese Beziehungen während des Entwurfsprozesses.

Im Zerstörungsreichen Modell werden folgende Beziehungen verwendet, die auf die Flächen des Raumprogramms angewendet werden:

Fläche $i$	-	innerhalb	-	Fläche $j$
Fläche $i$	-	schneidet nicht	-	Fläche $j$
Fläche $i$	-	berührt	-	Fläche $j$
Fläche $i$	-	außerhalb	-	Fläche $j$
Fläche $i$	-	auf Innenkante	-	Fläche $j$
Fläche $i$	-	berührt nicht	-	Fläche $j$

Durch diese Beziehungen ist es möglich, fast jede Grundrisskonstellation zu beschreiben. Fläche  $i, j$  sind dabei Variablen, die mit den Werten jeder anderen Fläche ersetzt werden können. Die Beschreibung erfolgt, in Anlehnung an eine gebräuchliche Vorgehensweise des Architekten, mit Hilfe von Bubble-Diagrammen.

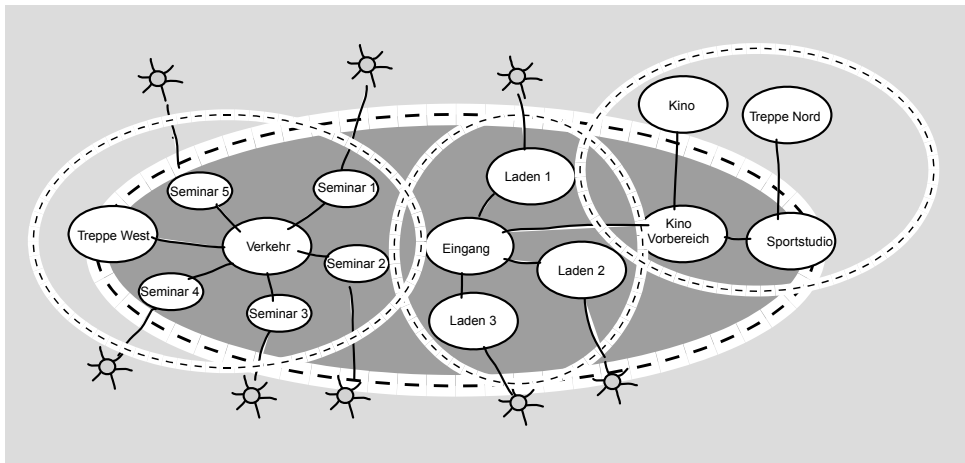


Abb. 5.1 Bubble-Diagramm eines Grundrisses

**ZA**

Das Zerstörungsarme Modell benutzt sowohl Daten über Flächen, die dem Raumprogramm entnommen werden, als auch Daten über vorhandene Flächen des existenten Gebäudegrundrisses.

Im Zerstörungsarmen Modell wird folgende Beziehung verwendet, die auf die Flächen des existenten Gebäudegrundrisses angewendet wird:

Fläche  $i$  - berührt - Fläche  $j$

Durch diese Beziehung wird die existente Grundrisskonstellation beschrieben. Die Beschreibung erfolgt mit Hilfe eines Graphen. Flächen werden in Form von Knoten dargestellt, Beziehungen zwischen Flächen in Form von Kanten.

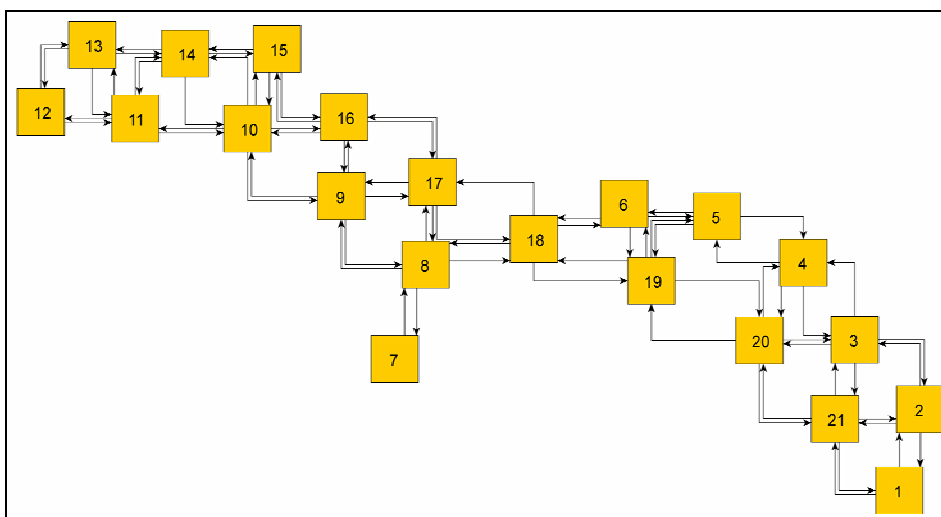


Abb. 5.2 Graphendarstellung

## 5.2 Geometriemodell

In der Architektur nimmt die 2-dimensionale Repräsentation eines Planungssachverhaltes eine gewichtige Rolle ein. Obwohl seit mehr als einem Jahrzehnt die Verwendung 3-dimensionaler geometrischer Modelle propagiert wird, ist festzuhalten, dass eine Vielzahl von Architekten aus vielfältigen Gründen nach wie vor 2-dimensionale Abbildungen den 3-dimensionalen vorziehen. So ist dieser Umstand u.a. der Tatsache geschuldet, dass Architekten von jeher ein ausgeprägtes räumliches Vorstellungsvermögen besitzen. Die Abstraktion eines räumlichen Objektes (Bauwerk) auf eine oder mehrere 2-dimensionale Abbildungen (Grundrisse, Schnitte, Ansichten) trägt in entscheidendem Maße zu einer Vereinfachung und einem besseren Verständnis bestimmter Planungssachverhalte bei. Des Weiteren steht der Nutzen 3-dimensionaler Modelle häufig nicht in vertretbarem Aufwand zu den entstehenden Kosten (Zeitaufwand der Erstellung).

Fläche	Fläche, Mathematik: zweidimensionales geometrisches Gebiet, Gesamtheit von Punkten, die sich in zwei (veränderlichen) Richtungen erstreckt, z. B. Ebene. [Web_Brockhaus, 2006]
--------	--

Die flächenhafte, 2-dimensionale Darstellung der Realität ist in der Architektur ein gebräuchliches Ausdrucksmittel in Bauzeichnungen. Im Geometriemodell repräsentieren daher 2-dimensionale Abbildungen von Flächen die Grundflächen von Räumen oder Nutzungseinheiten (z.B. HNF 1 gem. DIN 277 T1).

### 5.2.1 Komponenten des Geometriemodells

Die nachfolgend verwendeten Geometriemodelle basieren auf 2-dimensionalen Abbildungen von Grundrissen, die in ihrer Form abstrahiert und in Flächen (Units) zerlegt werden. Die Units können in ihrer geometrischen Ausprägung und Position variabel oder konstant sein. Folgende Formen von Units werden unterschieden:

Rechteckige Units	Parallelogramme mit 4 Rechten Winkeln und paralleler Ausrichtung der Kanten zu den geographischen Himmelsrichtungen (z.B. Rechtecke, Quadrate)
Rechtwinklige Units	Flächen mit mehr als 4 Kanten, jedoch ausschließlich Rechten Winkeln und paralleler Ausrichtung der Kanten zu den geographischen Himmelsrichtungen (z.B. L- und T-Form)
Nicht-Rechtwinklige Units	Flächen mit anderen als Rechten Winkeln und beliebiger Kantenzahl (z.B. schiefwinklige Flächen, Kreise, Ellipsen)

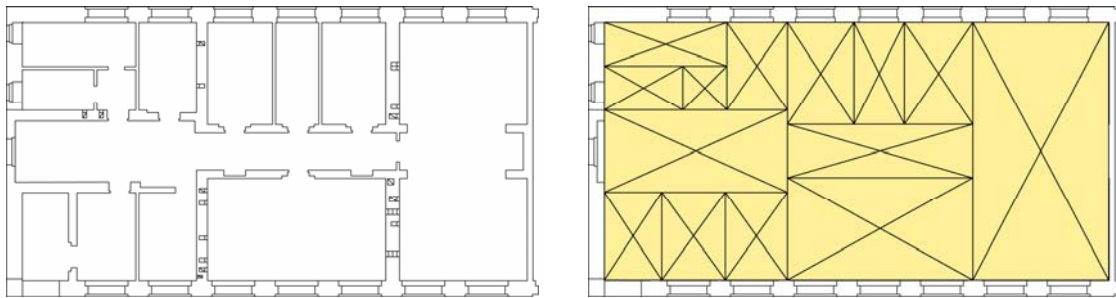


Abb. 5.3 Grundrissabstraktion – Zerlegung in rechteckige Units

### Units



Im ZR-Modell ist eine Unit die geometrische Repräsentation eines rechteckigen Raumes in Form einer rechteckigen Fläche. Sie kann durch die Zuweisung von Nebenbedingungen zu anderen Units in Beziehung gesetzt werden. Durch Verwendung des regelbasierten parametrisch-assoziativen Paradigmas (vgl. Kap. 4.4) kann auf jeden (freien) Parameter einer Unit explizit zugegriffen werden. Diese freien Parameter werden im Optimierungsprozess (vgl. Kap. 5.3) gezielt eingesetzt, um sie mit Hilfe von Nebenbedingungen an vorgegebene Zielgrößen anzunähern. Jede Unit  $i$  besitzt einen Referenzpunkt  $P(x_i, y_i)$  und zwei  $(\Delta x_i, \Delta y_i)$  oder vier  $(N_i, S_i, E_i, W_i)$  mit diesem Referenzpunkt verknüpfte Abstandsvariablen, aus welchen die Größe der Unit und die Position der Kanten (Wände) resultieren. Der Referenzpunkt einer Unit liegt bei Verwendung von zwei verknüpften Abstandsvariablen in der linken unteren Ecke einer Unit. Der Referenzpunkt einer Unit mit vier Abstandsvariablen kann vom Optimierungsalgorithmus arbiträr angeordnet werden. Durch Änderung von Abszisse oder Ordinate des Referenzpunktes wird eine Unit innerhalb des geometrischen Modells verschoben. Die Änderung der Abstandsvariablen hat eine Änderung der Größe zur Folge. Zur Erzeugung rechtwinkliger Flächen können Units gleicher Nutzungsart durch die Verwendung von Force Connection Constraints (FConn) miteinander kombiniert werden (vgl. Kap. 5.3.2.3).

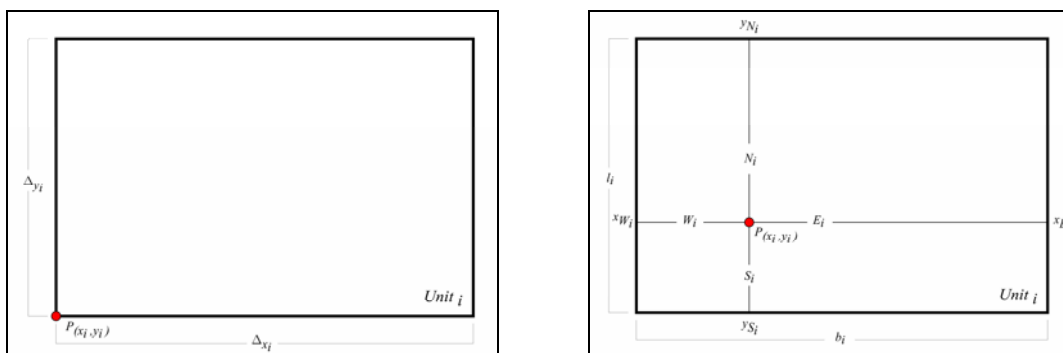


Abb. 5.4 Geometrische Definition einer Unit mit zwei bzw. vier Abstandsvariablen

ZA

Das Zerstörungsarme Modell orientiert sich an der geometrischen Ausprägung existenter Raumbegrenzungen oder der Lage von Systemlinien, wie denen des statischen Systems oder Rastermaßen von Bauteilkomponenten. Diese Abstraktion der Grundrisse gewährleistet, dass zerstörungsarm mit dem Gebäudebestand gearbeitet werden kann. Units des ZA-Modells sind unbeschränkt in ihrer geometrischen Ausprägung. In Form von Knoten eines Graphen können sie jede, auch nicht-rechtwinklige Fläche repräsentieren. Bei ZA-Modellen ist es nicht notwendig, Referenzpunkte oder Abstandsvariablen zu bestimmen, da die Position der Units konstant ist.

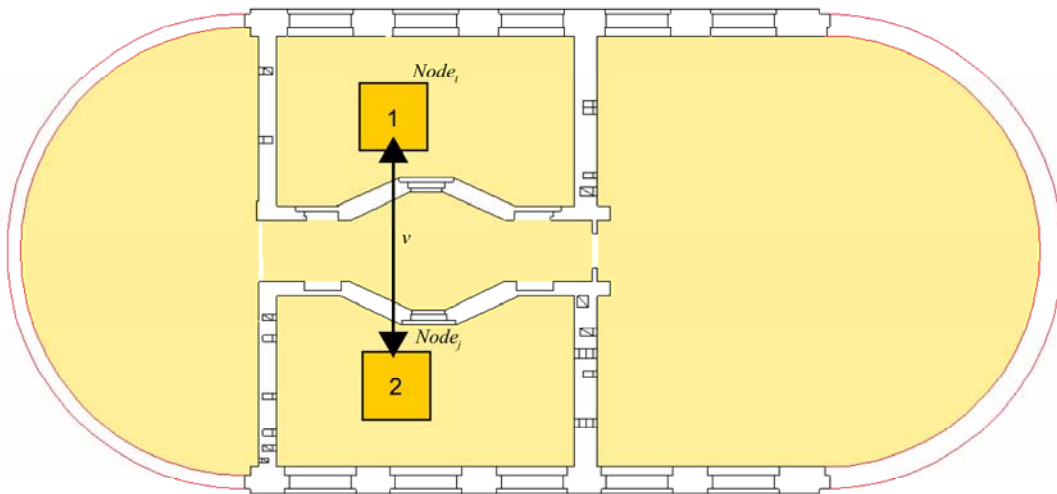


Abb. 5.5 Definition nicht-rechtwinkliger Units

*Definition: Eine Unit ist die geometrische Abstraktion der Fläche eines Raumes oder einer durch Systemlinien bildbaren Raumeinheit. Units sind in ihrer Position und Größe (ZR) oder Größe und Gruppenzugehörigkeit (ZA) durch Variablen definiert.*

### Grundunit

ZR

Diese Unit ist die geometrische Abstraktion des Gebäudeumrisses. Die Grundunit  $R$  ist gemäß der mathematischen Definition einer Unit in ihrer geometrischen Ausprägung ebenfalls auf eine Rechteckform begrenzt und parametrisiert. Die Parameter der Grundunit sind auf Grund ihrer Unveränderlichkeit als Konstanten definiert. Die Beschränkung der Units auf rechteckige For-

men hat zur Folge, dass ausschließlich Grundrisskonfigurationen untersucht werden können, deren Umriss ein Rechteck darstellt. Diese Konstellation findet sich in der Realität jedoch nur in wenigen Fällen, so dass komplexere Grundrisse in eine Vielzahl kleinerer rechteckiger Grundunits aufgeteilt und unabhängig voneinander untersucht werden müssten. Diese Vorgehensweise ist dann anwendbar, wenn voneinander unabhängige Nutzungseinheiten in diese rechteckigen Grundunits eingearbeitet werden können. Im Falle voneinander abhängiger Raumeinheiten führt eine derartige Unterteilung jedoch nur in den seltensten Fällen zu plausiblen Lösungen. Kapitel 5.3.2.3 zeigt, dass durch die Einführung so genannter Void-Units und die Anwendung von Constraint-Kombinationen auch voneinander abhängige Nutzungseinheiten in beliebigen Grundrisskonstellationen, die auf rechtwinklige Geometrien zurückführbar sind, untersucht werden können. Diese Constraint-Kombinationen in Verbindung mit Void-Units sind eine entscheidende Erweiterung von Modellen, die mit rechteckigen Geometrien arbeiten. Grundunits des ZR-Modells können dadurch rechteckige oder rechtwinklige Formen abbilden.



Die Grundunit des ZA-Modells unterliegt keinerlei geometrischer Beschränkung, d.h. sie kann jede, auch nicht-rechteckige oder nicht-rechtwinklige, geometrische Form abbilden.

#### Void-Units



Da rechtwinklige Grundrisse mehr als vier Kanten aufweisen, ist der Bezug der Constraints zu den vier Himmelsrichtungen nicht mehr eindeutig definiert. Ein derartiger Grundriss besitzt daher in jedem Fall mindestens zwei Kanten, mit gleicher geographischer Orientierung jedoch ungleichen Abszissen oder Ordinaten. Dieser Umstand führt dazu, dass Constraints, die lediglich einen einzigen Koordinatenbezug zu der jeweiligen Himmelsrichtung aufweisen können, nicht anwendbar sind. Die Erweiterung des geometrischen Modells auf rechtwinklige Grundunits wird daher aus der Differenz einer imaginären rechteckigen Grundunit, mit der darin enthaltenen rechtwinkligen Grundunit erreicht. Die verbleibende Differenzfläche, die durch die Verwendung von Prohibit Intersection Constraints (vgl. Kap. 5.3.2.3) von keiner anderen Unit eingenommen werden kann, wird in rechteckige Void-Units unterteilt, die in Position und Größe unveränderlich sind (Abb. 5.6). Diese konstanten Parameter der Void-Units können durch logische Constraint-Kombinationen (Constraint 1 oder Constraint 2) in die Optimierungsrechnung integriert werden. Die Anordnung einer Unit an einer Außenkante der Grundunit wird durch die Kombination eines Force To Border Constraints (F2Border) mit einem FConn Constraint ermöglicht. Die anzuordnende Unit liegt durch das F2Border Constraint entweder an einer regulären Außenkante, oder durch das FConn Constraint an einer Innenkante der Void-Unit, welche im Fall der nicht rechteckigen Grundunit wiederum einer Außenkante entspricht.

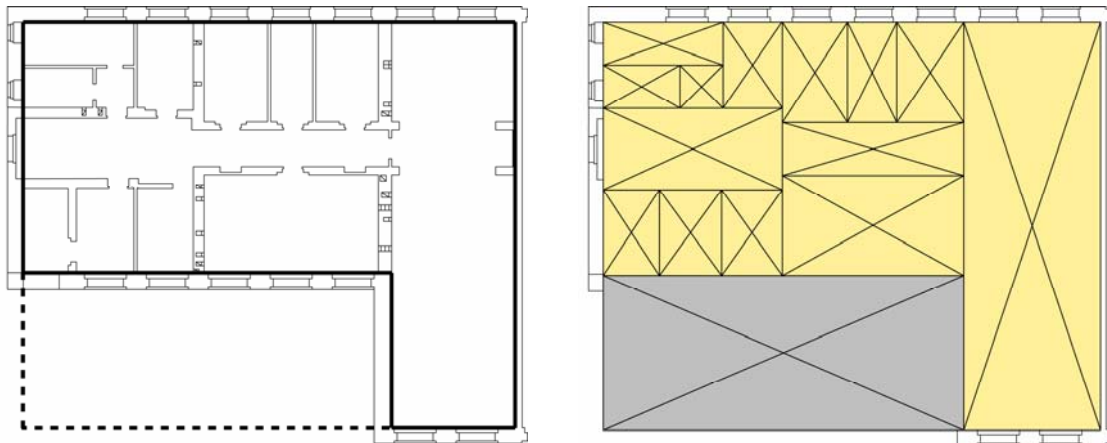


Abb. 5.6 Void-Unit (grau) in imaginärer rechteckiger Grundunit

**ZA**

Die Grundunit des ZA-Modells unterliegt keinerlei geometrischer Beschränkung. Void-Units sind daher nicht notwendig.

### Variablen

**ZR**

Zu den Variablen der Units zählen die Referenzpunkte  $P(x_i, y_i)$ , sowie die Abstände der Kanten vom Referenzpunkt  $(\Delta x_i, \Delta y_i)$  oder  $(N_i, S_i, E_i, W_i)$ . Aus der Änderung dieser Parameter resultiert eine Positions- oder Proportionsänderung der Unit. Parameter, welche während des Optimierungsprozesses verändert werden können, werden in Optimierungsmodellen gemeinhin als unabhängige Variablen oder „Design Variablen“ bezeichnet. Aus den Design Variablen können durch mathematische Operationen resultierende oder abhängige Variablen (Response Variablen), die als Parameter in Zielfunktionen oder Nebenbedingungen eingesetzt werden können, gebildet werden.

**ZA**

Das Optimierungsmodell erlaubt die Einführung parametrisierter Variablen in Form alphanumerischer Werte (z.B.: Material, Bezeichnung, Nutzungsart), welche auch im Optimierungsprozess eingesetzt werden können. Diese Parameter finden im Zerstörungssarmen Modell Verwendung. Folgende Variablen werden eingesetzt: Anzahl der existenten Units, bzw. Knoten ( $nbreOfNodes$ ), Anzahl der gesuchten Gruppen ( $nbreOfGroups$ ), Gruppenzugehörigkeit der Units

(*label*), Adjazenz zwischen Units (*A*), Größe der Units (*sizeNodes*), Mitgliederanzahl von Gruppen (*groups*), Größe von Gruppen (*sizeGroups*), Größe einer Unit innerhalb einer Gruppe (*sizeRooms*).

### Konstanten

#### ZR

Konstanten sind feststehende Parameter, deren Werte während einer Optimierungsrechnung unveränderlich bleiben. In der Regel brauchen Konstanten, die im Optimierungsprozess nicht verwendet werden, in der Problemdefinition auch nicht benannt zu werden. Zum besseren Verständnis der Optimierungsaufgabe ist ihre Nennung jedoch oft hilfreich. Die Position des Referenzpunktes  $P(x_R, y_R)$ , die Abstände der Kanten vom Referenzpunkt  $(l_R, b_R)$  der Grundunit und die Parameter der Void-Units sind Konstanten.

#### ZA

Beim ZA-Modell gibt es keine konstanten Parameter, die im Optimierungsprozess Verwendung finden. Auch die Anzahl der Units und die Adjazenz zwischen einzelnen Units sind keine durchgängigen Konstanten, da der Optimierungsprozess bei diesem Modell sich gerade dadurch auszeichnet, dass der Architekt diese Parameter zur besseren Zielerfüllung variieren kann. So können bspw. mehrere kleine Units zu einer größeren zusammengefasst werden, wenn davon ausgegangen werden kann, dass ihre ursprüngliche Größe keine vernünftige architektonische Lösung zulässt.

## 5.3 Optimierungsmodell

Optimierungsprobleme treten in vielen Bereichen des täglichen Lebens auf. Auf Flughäfen muss das Fluggeschehen vom Betreiber so koordiniert werden, dass eine möglichst hohe Auslastung unter Berücksichtigung der maximal möglichen Kapazität erreicht wird. In der Planung von Fahrstrecken für die Personenbeförderung müssen Haltepunkte und Verkehrswegenutzungen so geplant werden, dass bei geringstmöglichem Verbrauch an Kraftstoff, die maximal mögliche Anzahl an Personen zu einer optimalen Anzahl von Fahrzielen befördert werden kann. In der Güterproduktion müssen Maschinenzeiten durch Logistik-Manager derart organisiert werden, dass die Anforderungen der Kunden jederzeit befriedigt werden können, während der Gewinn des produzierenden Unternehmens maximiert und die Produktions- und Lagerhaltungskosten minimiert werden. Auf Großbaustellen muss der Einsatz einer Vielzahl von Handwerkern in Form von Einsatzplänen koordiniert werden, so dass keine Leerlaufzeiten entstehen und sich die für die Durchführung eines Arbeitsganges benötigten Ressourcen rechtzeitig am dafür optimalen Aufstellungsstandort befinden. In der Tragwerksplanung müssen Tragwerkskomponenten so gewählt werden, dass sie die auf das Bauwerk wirkenden Lasten unter Verwendung wirtschaftlich vertretbarer Querschnitte abtragen. Gleichzeitig müssen jedoch auch Faktoren wie Verfügbarkeit der Materialien, Folgekosten der gewählten Konstruktion oder ästhetische Erfordernisse



berücksichtigt werden. Allen vorgenannten Beispielen sind drei Faktoren gemeinsam [Bhatti, 2000]:

1. Es gibt ein übergeordnetes Ziel, welches erfüllt werden muss. Für den Flughafenbetreiber ist dies die Maximierung der Auslastung des Flugaufkommens, für den Verkehrsbetrieb die Minimierung des Kraftstoffverbrauchs. Für den Güterproduzenten, resp. den Logistik-Manager ist das übergeordnete Ziel die Maximierung des Gewinns. Da die Logistik nach Phasen des Güterflusses in Beschaffungslogistik, Produktionslogistik, Distributionslogistik und Retrologistik unterteilt werden kann [Mau, 2002], ist es durchaus denkbar, dass in einzelnen Logistikbereichen divergierende Ziele definiert werden. So ist die Zielsetzung der Produktionslogistik möglicherweise die Reduzierung der Lagerhaltungskosten. Diese Zielsetzung muss jedoch nicht zwangsläufig zum Erreichen des Gesamtzieles führen. Da Konstellationen denkbar sind, die eine extensive Lagerhaltung rechtfertigen könnten, würde die Reduzierung der Lagerhaltungskosten in diesem Fall unter Umständen das Erreichen des übergeordneten Zieles erschweren. Es ist möglich, sowohl die Maximierung des Gewinns als auch die Minimierung der Lagerhaltungskosten als Zielvorgabe zu definieren. In diesem Fall spricht man von Optimierungsproblemen mit mehrfacher Zielsetzung (Multi-Objective Optimization).
2. Zulässige Lösungen zur Zielerreichung unterliegen bestimmten Bedingungen. So darf der Flughafen beispielsweise nur zu bestimmten Tages- und Nachtzeiten angefliegen werden. Durch die Verkehrswegeplanung des Personenbeförderers dürfen weder Fußgängerzonen befahren noch Einbahnstrassen in der falschen Richtung benutzt werden. Der Güterproduzent kann nicht mehr produzieren, als die vollständige Auslastung seiner Maschinen und Mitarbeiter zulässt. Die Erreichung eines Zieles für ein Optimierungsproblem wird demnach durch die Festlegung von Nebenbedingungen restringiert.
3. Es gibt unterschiedliche Möglichkeiten ein Ziel unter Einhaltung der Nebenbedingungen zu erreichen. So kann der Flughafenbetreiber die Flughafensteuern senken, die Standgebühren für Flugzeuge verringern oder sein Angebot auf die Beförderung von Personen und Frachtgütern ausdehnen. Der Verkehrsbetrieb könnte den Kraftstoffverbrauch seiner Fahrzeuge senken, indem er unwirtschaftliche Strecken abbaut oder veraltete Fahrzeuge durch neue ersetzt. Faktoren, welche die Erfüllung der Nebenbedingungen und Ziele beeinflussen, werden Optimierungs- oder Design-Variablen genannt. Variablen, die keinen Einfluss auf Nebenbedingungen oder Optimierungsziele haben, spielen bei der Problemlösung keine Rolle und werden in der Regel im Optimierungsmodell nicht benannt.

### 5.3.1 Standardform eines Optimierungsmodells

In der mathematischen Literatur, insbesondere im Bereich des Operations Research, ist es üblich, von Problemen (Transportprobleme, Verpackungsprobleme, Standortprobleme) statt von Modellen zu sprechen. Optimierungsmodelle werden daher in der Regel als Optimierungsprobleme bezeichnet. In den nachfolgenden Kapiteln beziehen sich die beschriebenen Optimierungsmodelle auf die mathematisch modellhafte Umsetzung eines architektonischen Sachverhaltes, i.e. die formale Beschreibung von Grundrissen. Aus diesem Grund werden die Begriffe „Modell“ und „Optimierungsmodell“ im Sinn der nachfolgenden Definitionen verwendet.

*Definition:* „Ein Modell ist eine zweckorientierte, relationseindeutige Abbildung der Realität.“ [Gohout, 2004]

*Definition:* „Ein Optimierungsmodell ist eine formale Darstellung eines Entscheidungs- oder Planungsproblems, dass in seiner einfachsten Form mindestens eine Alternativenmenge und eine diese bewertende Zielfunktion enthält.“ [Domschke und Drexl, 2005, S. 3].

Der Begriff „Optimierungsproblem“ wird im Sinne der Beschreibung der zu optimierenden Problemstellung angewandt und bezieht sich in der Regel auf die Formulierung der Zielfunktion.

Optimierungsmodelle bestehen in ihrer Standardform aus einer gegebenen Anzahl konstanter und variabler Werte, einer oder mehrerer Zielfunktionen sowie aus einer veränderlichen Anzahl an Nebenbedingungen. Die Standardform eines allgemeinen Optimierungsmodells lässt sich wie folgt darstellen [Domschke und Drexl, 2005]:

Maximiere oder minimiere die Zielfunktion:

$$z = F(x) \quad (1)$$

unter Einhaltung der Nebenbedingungen:

$$g_i(x) \begin{cases} \geq \\ = \\ \leq \end{cases} 0 \quad \text{für } i = 1, \dots, m \quad (2)$$

$$x \in W_1 \times W_2 \times \dots \times W_n, \quad W_j \in \{\mathbb{R}_+, \mathbb{Z}_+, \mathbb{B}\}, \quad j = 1, \dots, n \quad (3)$$

Die Symbole haben folgende Bedeutung:

$x$  ein Variablenvektor mit  $n$  Komponenten  $x_1, \dots, x_n$

$F(x)$  eine Zielfunktion

$x_j \in \mathbb{R}_+$  Nichtnegativitätsbedingung

$x_j \in \mathbb{Z}_+$  Ganzzahligkeitsbedingung

$x_j \in \mathbb{B}$  Binärbedingung (binäre Variable)

### Zielfunktionen (Objectives)

(1) entspricht der zu maximierenden oder minimierenden Zielfunktion. Verallgemeinernd spricht man bei zu maximierenden Zielfunktionen von Nutzen (Erträge, Gewinn, Auslastung) und bei zu minimierenden Zielfunktionen von Kosten (Verbrauch, Fahrstrecke, Projektdauer). Häufig ist es schwierig, ein Optimierungsproblem so zu formulieren, dass eine einzige Zielfunktion ausreichend ist, um zu einer optimalen Gesamtlösung des Problems zu gelangen. Die Ziele eines derartigen Problems mit mehrfacher Zielsetzung können zueinander komplementär, konkurrierend (konträr) oder neutral sein. Im Fall komplementärer Ziele entsteht kein Zielkonflikt, da die Menge der zulässigen Lösungen mindestens einen Eckpunkt enthält, der für alle Zielfunktionen ein Optimum darstellt. In diesem Fall spricht man von einer perfekten Lösung. Im Fall konkurrierender Ziele tritt ein Zielkonflikt auf, da sich mit der Verbesserung des Zielerreichungsgrades eines Zieles derjenige des anderen Zieles verschlechtert. Eine mögliche Strategie zur Lösung eines solchen Konfliktes ist die Zuordnung von Gewichten ( $w_1, w_2, \dots$ ) zu den jeweiligen Zielfunktionen und die Bildung einer gemeinsamen Funktion als gewichtete Summe der Ausgangsfunktionen:

$$f(x) = w_1 f_1(x) + w_2 f_2(x) + \dots \quad (4)$$

Eine weitere Möglichkeit ist die Auswahl des wichtigsten Zieles als einzige Zielfunktion und die Festlegung der restlichen Zielfunktionen als Nebenbedingungen mit entsprechenden Oberen und Unteren Schranken. Zueinander neutrale Zielfunktionen beeinflussen sich in ihrer Wirkungsweise nicht. Da nicht alle Optimierungsprogramme die Lösung mehrerer Zielfunktionen unterstützen, wird häufig versucht, das Problem in mehreren Optimierungsrechnungen unter jeweiliger Verwendung neuer Zielfunktionen zu lösen. Diese Vorgehensweise ist jedoch in der architektonischen Grundrissplanung wenig Erfolg versprechend, da die Beibehaltung der bereits erreichten Optimierungsergebnisse in neuerlichen Berechnungen sowie die Erzielung eines optimalen Ergebnisses unter Verwendung neuer Zielfunktionen nicht gewährleistet ist. Die Verwendung eines bereits vorhandenen Optimierungsergebnisses in einer neuerlichen Berechnung ohne Verwendung einer Zielfunktion kann hingegen sinnvoll sein, da basierend auf einer optimalen Lösung Varianten entwickelt werden können, die allesamt den optimalen Wert der bereits berechneten Zielfunktion aufweisen (Constraint Satisfaction). Ebenso kann die Verwendung divergierender Zielrichtungen (gleichzeitiges Maximum und Minimum) bei mehreren Zielsetzungen problematisch sein, weil den Solvern der Optimierungsprogramme oftmals übergeordnete Optimierungsziele (entweder Maximum oder Minimum) vorgegeben werden müssen. Das Problem kann dadurch gelöst werden, dass in der linearen Programmierung eine zu minimierende Zielfunktion  $z = F(x)$  durch die zu maximierende Zielfunktion  $-z = -F(x)$  ersetzt wird.

### Nebenbedingungen (Constraints)

(2) repräsentiert ein System von  $m$  Gleichungen und/oder Ungleichungen, die als Restriktionen den Lösungsraum einer Optimierungsaufgabe einschränken. Nebenbedingungen können durch die Vergabe Oberer und Unterer Schranken (Upper- and Lower Bounds) eingeschränkt werden:

$$x_{iL} \leq x_i \leq x_{iU} \quad i = 1, 2, \dots, n \quad (5)$$

Tatsächlich sind Obere und Untere Schranken Ungleichungssysteme, die jedoch auf Grund ihrer besonderen Form häufig von den Nebenbedingungen getrennt werden. Bestimmte numerische Optimierungsalgorithmen machen von dieser besonderen Form Gebrauch und gestalten den Berechnungsprozess dadurch effizienter. In einigen Fällen ist die Verwendung von Gleichungen in Nebenbedingungen problematisch. In diesem Fall kann ein Gleichungssystem der Form  $a_{i1}x_1 + \dots + a_{ip}x_p = b_i$  durch zwei Ungleichungen  $a_{i1}x_1 + \dots + a_{ip}x_p \leq b_i$  und  $-a_{i1}x_1 - \dots - a_{ip}x_p \leq -b_i$  ersetzt werden. Die Standardform eines Optimierungsproblems berücksichtigt lediglich „kleiner gleich“ als Ungleichungsform. Ungleichungstypen der Form „größer gleich“ können durch Multiplikation mit -1 auf beiden Seiten der Ungleichung in die „kleiner gleich“ Form konvertiert werden.

### Variablen

(3) definiert den Wertebereich der Design-Variablen. Diese können kontinuierliche, ganzzahlige oder binäre Werte annehmen. Es ist möglich, dass Variablen auch negative Werte annehmen. Ist die Negativität wie im Beispiel der in Kap. 6.1 beschriebenen Optimierungsmodelle ausgeschlossen, ist es sinnvoll, Nichtnegativitätsbedingungen einzuführen. Die Wahl der richtigen Variablentypen für das Optimierungsmodell ist von entscheidender Bedeutung für die entstehenden Berechnungskosten. Optimierungsprobleme unter alleiniger Verwendung von Integer-Variablen (Integer Programming - IP) sind sehr viel schwieriger zu lösen als lineare Optimierungsprobleme mit reellen Variablen. Die Kombination aus reellen und Integer-Variablen (Mixed Integer-Linear Programming - MILP) benutzt die gleiche Technologie wie die Integer-Programmierung und ist somit ebenfalls kostenintensiver.

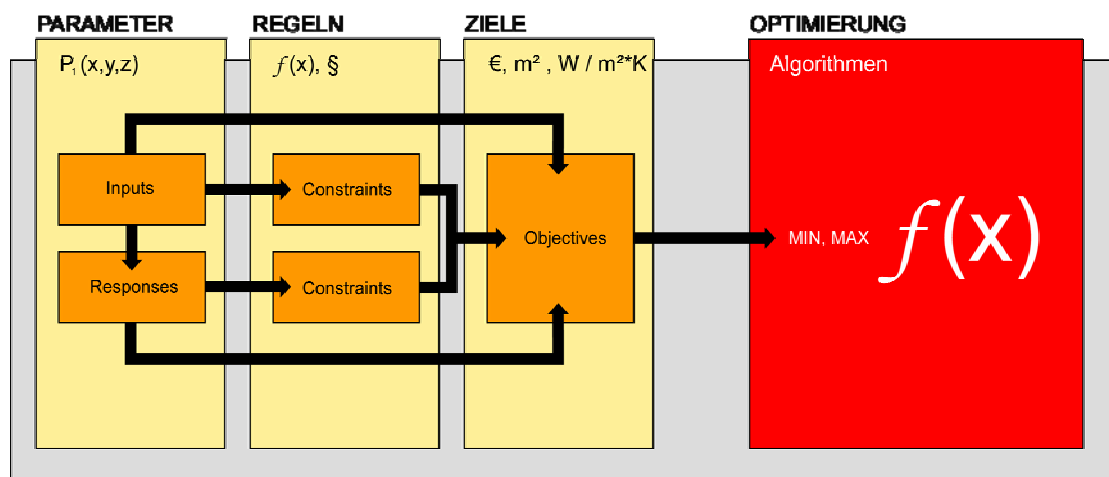


Abb. 5.7 Schematische Darstellung des Optimierungsmodells

### 5.3.2 Zerstörungsreiches Modell

ZR

Das Zerstörungsreiche Modell findet in der Revitalisierung Verwendung, wenn existente Freiräume innerhalb eines Gebäudes neu strukturiert und geordnet werden müssen. Dies ist auch dann der Fall, sofern eine existente Raumstruktur den Anforderungen an eine Nachnutzung nicht entspricht oder aus anderen Gründen baulich verändert werden muss. In diesem Fall werden durch den Abbruch von Wänden Freiräume geschaffen, die mit Hilfe dieser Modellbildung neu beplant werden können. Voraussetzung für diese Neuplanung ist ein existentes Raumprogramm, welches die neu zu schaffenden Flächen (Units) bezüglich der Nutzungsart und der Flächengröße beschreibt. Durch die Definition von Beziehungsparametern (Constraints) zwischen den Flächen, werden die Nebenbedingungen für den späteren Optimierungsprozess erzeugt. Um die Komplexität der auf eine Unit anzuwendenden Nebenbedingungen einzugrenzen, werden bei diesem Modell Units in Form rechteckiger geometrischer Flächen definiert. Die unterschiedlichen Ausprägungen des mathematischen Modells werden detailliert in Kapitel 6 (Mathematische Modellbildung) beschrieben. Da alle Nebenbedingungen und Zielfunktionen die geographischen Positionen (N, S, E, W) der Kanten einer Unit referenzieren, wird in Kapitel 5 zur Vereinfachung der Lesbarkeit eine Darstellung in Pseudo-Code gewählt. Mit Ausnahme von Kapitel 5.3.2.6 (Suchstrategien) ist die beschriebene Systemkonzeption unabhängig von der Verwendung bestimmter Programmiersprachen oder Optimierungsalgorithmen. Die Parameter  $(y_{Ni}, y_{Si}, x_{Ei}, x_{Wi})$  resp.  $(y_{Nj}, y_{Sj}, x_{Ej}, x_{Wj})$  bezeichnen die Kantenpositionen zweier Units  $i$  und  $j$ . Finden Nebenbedingungen ausschließlich Anwendung auf eine Grundunit  $R$ , werden die Konstanten  $(y_{NR}, y_{SR}, x_{ER}, x_{WR})$ , resp.  $l_R, b_R$  verwendet.

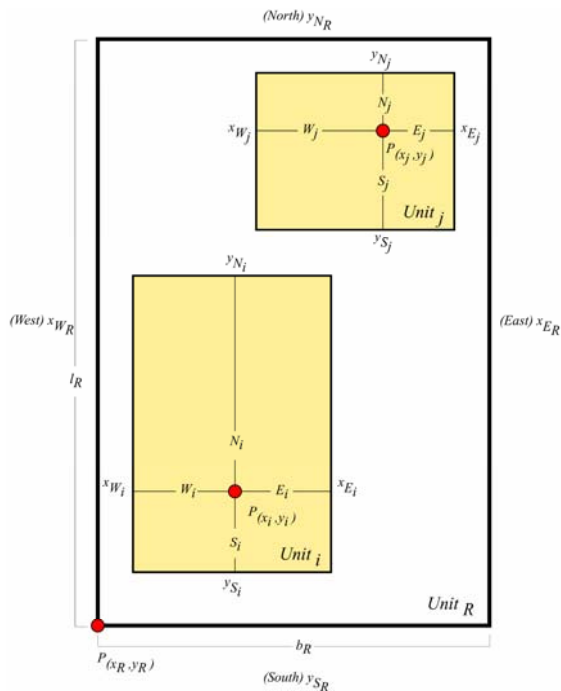


Abb. 5.8 Geometrische Definition der Units  $i$  und  $j$  innerhalb der Grundunit  $R$

### 5.3.2.1 Konstanten

Die Konstanten des Modells werden durch den Referenzpunkt der Grundunit und die Abstände der Kanten vom Referenzpunkt gebildet. Aus den Kantenabständen in N/S- bzw. E/W-Richtung lassen sich Länge und Breite der Grundunit sowie Flächeninhalt und Umfang errechnen. Diese konstanten Werte finden in Zielfunktionen und Nebenbedingungen Verwendung.

### 5.3.2.2 Variablen

Die Design-Variablen des Modells werden durch die Referenzpunkte der Units und die Abstände der Kanten von den Referenzpunkten gebildet. Aus den Kantenabständen in N/S- bzw. E/W-Richtung einer Unit können Länge und Breite sowie Flächeninhalt und Umfang errechnet werden. Die Position der E/W-Kanten wird durch die Summe resp. Differenz der Abszisse des Referenzpunktes und des Kantenabstandes E/W vom Referenzpunkt erzeugt. Gleiches gilt für die Ordinate des Referenzpunktes unter Verwendung des Kantenabstandes N/S für die Kantenposition in nördlicher und südlicher Richtung. Die Response-Variablen finden in Zielfunktionen und Nebenbedingungen Verwendung.

### 5.3.2.3 Nebenbedingungen

Das gewählte Optimierungsmodell verwendet die von Michalek [Michalek, 2001] entwickelten Nebenbedingungen (Constraints) „Force Inside“, „Prohibit Intersection“ und „Force To Edge“

(resp. Force To Border). Zur detaillierten Definition der Beziehungen zwischen zwei Units werden zusätzliche Constraints (Force Connection, Force Outside, Prohibit Connection, Design Constraints) und Constraint-Kombinationen eingeführt. Mit Hilfe dieser Constraints ist es möglich, die Position der Units gezielt zu beeinflussen. Um die Anzahl erzeugbarer Grundrissanordnungen zu erhöhen, ist es sinnvoll, selten Constraintgruppen zu verwenden, welche die Position der Units einschränken.

### Force Inside Constraints (FInside)

Force Inside Constraints werden verwendet, um eine Unit innerhalb einer anderen Unit zu platzieren. In der Revitalisierung kommt den Force Inside Constraints eine wichtige Aufgabe zu. Da das wesentliche Ziel der Optimierungsrechnung die Integration des neuen Raumprogramms innerhalb der existenten Grundrissfläche ist, wird mittels dieses Constraints versucht, alle Units innerhalb der Grundunit anzuordnen. Eine Unit  $i$  befindet sich genau dann innerhalb einer Unit  $j$ , wenn gleichzeitig alle nachfolgenden Bedingungen erfüllt sind:

$$(y_{Ni} \leq y_{Nj}) \text{ AND } (y_{Sj} \leq y_{Si}) \text{ AND } (x_{Ei} \leq x_{Ej}) \text{ AND } (x_{Wj} \leq x_{Wi}) \quad (6)$$

$$y_{Ni} < y_{Nj} \quad \text{Unit } i \text{ befindet sich südlich der nördlichen Innenkante von Unit } j \quad (7)$$

$$y_{Sj} < y_{Si} \quad \text{Unit } i \text{ befindet sich nördlich der südlichen Innenkante von Unit } j \quad (8)$$

$$x_{Ei} < x_{Ej} \quad \text{Unit } i \text{ befindet sich westlich der östlichen Innenkante von Unit } j \quad (9)$$

$$x_{Wj} < x_{Wi} \quad \text{Unit } i \text{ befindet sich östlich der westlichen Innenkante von Unit } j \quad (10)$$

Die Force Inside Constraints (6) sind vollständig erfüllt, wenn der größte Wert innerhalb der Argumentliste kleiner oder gleich Null ist. Ist der größte Wert der Argumentliste gleich Null, befindet sich Unit  $i$  innerhalb und gleichzeitig auf einer Innenkante von Unit  $j$  (vgl. Force To Border Constraint):

$$y_{Ni} = y_{Nj} \quad \text{Unit } i \text{ befindet sich auf der nördlichen Innenkante von Unit } j \quad (11)$$

$$y_{Sj} = y_{Si} \quad \text{Unit } i \text{ befindet sich auf der südlichen Innenkante von Unit } j \quad (12)$$

$$x_{Ei} = x_{Ej} \quad \text{Unit } i \text{ befindet sich auf der östlichen Innenkante von Unit } j \quad (13)$$

$$x_{Wj} = x_{Wi} \quad \text{Unit } i \text{ befindet sich auf der westlichen Innenkante von Unit } j \quad (14)$$

Das Constraint findet in der Regel Anwendung auf Units  $(i, j)$  innerhalb  $R$  und ist daher die Grundeinstellung für alle Units  $(i, j)$ . Dennoch ist es möglich, dass ein FInside Constraint auch unabhängig von der Grundunit  $R$  Verwendung finden kann. Das ist immer dann der Fall, wenn Units ineinander verschachtelt werden müssen und die Anordnung an einer bestimmten Innenkante der verschachtelnden Unit unmaßgeblich ist.

### Prohibit Intersection (PInter)

Den Prohibit Intersection Constraints kommt die Aufgabe zu, Überschneidungen zwischen den Units zu vermeiden. Zwei Units  $i$  und  $j$  überschneiden sich nicht, wenn Unit  $i$  außerhalb von Unit  $j$  liegt. PInter und FInside Constraints dürfen daher nicht gleichzeitig für zwei Units  $i$  und  $j$  verwendet werden. Eine Unit  $i$  liegt genau dann außerhalb einer Unit  $j$ , wenn mindestens eine der nachfolgenden Bedingungen erfüllt ist:

$$(y_{Ni} \leq y_{Sj}) \text{OR} (y_{Nj} \leq y_{Si}) \text{OR} (x_{Ei} \leq x_{Wj}) \text{OR} (x_{Ej} \leq x_{Wi}) \quad (15)$$

$$y_{Ni} < y_{Sj} \quad \text{Unit } i \text{ befindet sich südlich von Unit } j \quad (16)$$

$$y_{Nj} < y_{Si} \quad \text{Unit } i \text{ befindet sich nördlich von Unit } j \quad (17)$$

$$x_{Ei} < x_{Wj} \quad \text{Unit } i \text{ befindet sich westlich von Unit } j \quad (18)$$

$$x_{Ej} < x_{Wi} \quad \text{Unit } i \text{ befindet sich östlich von Unit } j \quad (19)$$

Die Prohibit Intersection Constraints sind vollständig erfüllt, wenn der kleinste Wert der Argumentliste kleiner oder gleich Null ist. Ist der kleinste Wert der Argumentliste gleich Null, befindet sich Unit  $i$  auf der Aussenkante von Unit  $j$  (vgl. Force Outside Constraint):

$$y_{Ni} = y_{Sj} \quad \text{Unit } i \text{ befindet sich auf der südlichen Aussenkante von Unit } j \quad (20)$$

$$y_{Nj} = y_{Si} \quad \text{Unit } i \text{ befindet sich auf der nördlichen Aussenkante von Unit } j \quad (21)$$

$$x_{Ei} = x_{Wj} \quad \text{Unit } i \text{ befindet sich auf der westlichen Aussenkante von Unit } j \quad (22)$$

$$x_{Ej} = x_{Wi} \quad \text{Unit } i \text{ befindet sich auf der östlichen Aussenkante von Unit } j \quad (23)$$

Entgegen der Vorgehensweise von Michalek [Michalek, 2001] wird zu Gunsten eindeutiger Grundrisslösungen auf die Verwendung eines zusätzlichen Constraints verzichtet, welches Überschneidungen zwischen Units ermöglicht. Mit Hilfe dieser Überschneidungen wäre es



theoretisch möglich, komplexere, rechtwinklige Anordnungen von Units zu erzielen. Die sich überschneidenden Formen führen jedoch zu nicht eindeutigen Raumanordnungen. In der vorliegenden Arbeit können rechtwinklige Formen durch die Verwendung der entwickelten Void-Units realisiert werden. Das Prohibit Intersection Constraint findet in der Regel Anwendung auf Units  $(i, j)$  innerhalb  $R$  und ist daher die Grundeinstellung für alle Units  $(i, j)$ . Es darf nicht verwendet werden, wenn zwei Units ineinander verschachtelt werden müssen.

### Force Connection (FConn)

Mit Hilfe dieser Constraintgruppe wird die Berührung zweier Units  $i$  und  $j$  erzwungen. Berührungen sind dann von besonderer Wichtigkeit, wenn zwei Flächen zwingend nebeneinander liegen müssen. Eine solche Bedingung liegt in der Revitalisierung insbesondere dann vor, wenn versucht wird die vertikale Erschließungsstruktur und die Lage der Nebennutzflächen konstant zu halten (vgl. Kap. 2.2.2). Mit Hilfe des FConn Constraints werden dabei neue Erschließungsflächen an die existente vertikale Erschließungsstruktur angebunden. Die Lage neu zu schaffender Nebennutzflächen wird, auf Grund der vorteilhaften Nutzung vorhandener Schächte zur Installationsführung, ebenfalls mit Hilfe des FConn Constraints auf bestehende Bereiche dieser Flächenart konzentriert.

Eine Unit  $i$  berührt eine Unit  $j$ , wenn  $i$  und  $j$  eine gemeinsame Kante geographisch konträrer Himmelsrichtung besitzen und gleichzeitig eine der Kanten der verbleibenden Himmelsrichtungen von  $i$  zwischen den Kanten der äquivalenten Himmelsrichtungen von  $j$  liegt. Eine Unit  $i$  berührt auch dann eine Unit  $j$ , wenn  $i$  und  $j$  über mindestens einen gemeinsamen Punkt verfügen, welcher durch die Berührung von Kanten konträrer geographischer Himmelsrichtungen gebildet wird. Die Berührung zeichnet sich in der Regel jedoch dadurch aus, dass eine Erschließung zwischen den sich berührenden Units möglich ist. In diesem Fall wird das FConn Constraint entweder durch einen Faktor  $\delta$  oder mit Oberen und Unteren Schranken (Upper- and Lower Bounds) versehen, welche die Länge der Berührungskante definieren. FConn Constraints können nur auf Units angewandt werden, die nicht mit Hilfe des FInside Constraints ineinander verschachtelt sind. Für den Fall der Berührung zweier Units an Kanten geographisch äquivalenter Himmelsrichtungen, wird das Force To Border Constraint verwendet.

Eine Unit  $i$  berührt eine Unit  $j$ , wenn mindestens eine Zweier-Bedingung erfüllt ist:

$$\begin{aligned} & (y_{Nj} = y_{Si}) \text{ OR } (y_{Sj} = y_{Ni}) \text{ AND } \min(x_{Ej} - x_{Wi}, x_{Ei} - x_{Wj}) \geq 0 \\ & \text{OR} \\ & (x_{Ej} = x_{Wi}) \text{ OR } (x_{Wj} = x_{Ei}) \text{ AND } \min(y_{Nj} - y_{Si}, y_{Ni} - y_{Sj}) \geq 0 \end{aligned} \tag{24}$$

Unit  $i$  befindet sich nördlich und berührt Unit  $j$ , wenn das nachfolgende Constraintpaar erfüllt ist:

$$FConn N1(i, j): y_{Nj} - y_{Si} = 0 \tag{25}$$

$$FConn N2(i, j) : \min(x_{Ej} - x_{Wi}; x_{Ei} - x_{Wj}) \geq 0 \quad (26)$$

Unit  $i$  befindet sich südlich und berührt Unit  $j$ , wenn das nachfolgende Constraintpaar erfüllt ist:

$$FConn S1(i, j) : y_{Sj} - y_{Ni} = 0 \quad (27)$$

$$FConn S2(i, j) : \min(x_{Ej} - x_{Wi}; x_{Ei} - x_{Wj}) \geq 0 \quad (28)$$

Unit  $i$  befindet sich östlich und berührt Unit  $j$ , wenn das nachfolgende Constraintpaar erfüllt ist:

$$FConn E1(i, j) : x_{Ej} - x_{Wi} = 0 \quad (29)$$

$$FConn E2(i, j) : \min(y_{Nj} - y_{Si}; y_{Ni} - y_{Sj}) \geq 0 \quad (30)$$

Unit  $i$  befindet sich westlich und berührt Unit  $j$ , wenn das nachfolgende Constraintpaar erfüllt ist:

$$FConn W1(i, j) : x_{Wj} - x_{Ei} = 0 \quad (31)$$

$$FConn W2(i, j) : \min(y_{Nj} - y_{Si}; y_{Ni} - y_{Sj}) \geq 0 \quad (32)$$

### Force Outside (FOutside)

Force Outside Constraints verschieben Units auf eine der vier Außenseiten einer anderen Unit. Diese Constraints werden auf die Kombination einer Unit  $i, j$  mit der Grundunit  $R$  angewandt, wenn das Flächenprogramm das Flächenangebot des zu revitalisierenden Gebäudes überschreitet. In diesem Fall können einzelne Units mit Hilfe des FOutside Constraints außerhalb des Gebäudes angeordnet werden. Das FOutside Constraint kann somit als Anbauoption charakterisiert werden. Das FOutside Constraint darf nicht gleichzeitig auf zwei Units angewendet werden, auf welche ein FInside Constraint angewendet wird. Da mehr als eine Unit angebaut werden kann, müssen die Bedingungen der FOutside Constraints kleiner gleich dem Wert der entsprechenden Kante von  $R$  sein. Das FOutside Constraint entspricht in seinem Aufbau dem PInter Constraint, wird jedoch explizit definiert, da es ausschließlich auf Kombinationen mit der Grundeinheit  $R$  Anwendung findet:

$$(y_{Ni} \leq y_{SR}) OR (y_{NR} \leq y_{Si}) OR (x_{Ei} \leq x_{WR}) OR (x_{ER} \leq x_{Wi}) \quad (33)$$

$$y_{Ni} < y_{SR} \quad \text{Unit } i \text{ befindet sich südlich von Unit } R \quad (34)$$

$$y_{NR} < y_{Si} \quad \text{Unit } i \text{ befindet sich nördlich von Unit } R \quad (35)$$

$$x_{Ei} < x_{WR} \quad \text{Unit } i \text{ befindet sich westlich von Unit } R \quad (36)$$

$$x_{ER} < x_{Wi} \quad \text{Unit } i \text{ befindet sich östlich von Unit } R \quad (37)$$

Für die Positionierung an einer expliziten Außenkante werden die folgenden Bedingungen angewendet:

$$y_{Ni} = y_{SR} \quad \text{Unit } i \text{ befindet sich auf der südlichen Außenkante von Unit } R \quad (38)$$

$$y_{NR} = y_{Si} \quad \text{Unit } i \text{ befindet sich auf der nördlichen Außenkante von Unit } R \quad (39)$$

$$x_{Ei} = x_{WR} \quad \text{Unit } i \text{ befindet sich auf der westlichen Außenkante von Unit } R \quad (40)$$

$$x_{ER} = x_{Wi} \quad \text{Unit } i \text{ befindet sich auf der östlichen Außenkante von Unit } R \quad (41)$$

Wie beim FConn Constraint sind auch beim FOutside Constraint zwei Bedingungen erforderlich, um zwei Units miteinander zu verbinden. Eine Kombination von zwei Bedingungen wie bei FConn schließt jedoch bei FOutside die Möglichkeit aus, eine Unit außerhalb der Grundunit zu platzieren, ohne die Grundunit  $R$  zu berühren. Diese Option muss jedoch ermöglicht werden, da beispielsweise zwei außen liegende Units  $i$  und  $j$  so angeordnet sein könnten, dass Unit  $i$  durch Unit  $j$  erschlossen wird und demnach keine Berührungskante mit  $R$  notwendig ist. Das FConn Constraint würde jedoch versuchen, alle Units direkt an der Außenkante von  $R$  anzuordnen. Die Anordnung einer außen liegenden Unit  $i$  wird durch ein FConn Constraint zu einer innerhalb der Grundunit  $R$  liegenden Unit  $j$  gewährleistet. Weitere außen liegende Units ( $m, n, o, \dots$ ) werden durch ein FConn Constraint zu  $i$  zusammenhängend angeordnet.

### Force To Border (F2Border)

Die Constraints dieser Gruppe werden eingesetzt, um eine Unit gezielt an der Innenkante einer anderen Unit zu positionieren. Dies ist zum Beispiel der Fall, wenn Flächen natürlich belichtet werden müssen oder eine direkte Verbindung nach außen erforderlich ist. Eine Unit  $i$  liegt innen an einer Innenkante von Unit  $j$ , wenn  $i$  und  $j$  eine gemeinsame Kante geographisch äquivalenter Himmelsrichtung haben und  $i$  gleichzeitig innerhalb von  $j$  liegt (FInside). In der Regel wird das Force To Border Constraint nur auf Units  $i, j$  in Verbindung mit der Grundunit  $R$  angewendet. Da jedoch die Lage einer Unit  $i$  innerhalb einer Unit  $j$  ebenfalls möglich ist (z.B.:

WC-Zellen innen an einer bestimmten Außenwand eines Toilettenraumes), wird hier die Syntax für eine Verbindung der Units  $i, j$  verwendet. Wenn die Positionierung ohne Benennung der Himmelsrichtung erfolgt, kann das Force To Border Constraint folgendermaßen zusammengefasst werden:

$$(y_{Ni} = y_{Nj})OR(y_{Si} = y_{Sj})OR(x_{Ei} = x_{Ej})OR(x_{Wi} = x_{Wj}) \quad (42)$$

Für die gezielte Positionierung an einer Innenkante werden die folgenden Bedingungen angewendet:

$$y_{Ni} = y_{Nj} \quad \text{Unit } i \text{ befindet sich auf der nördlichen Innenkante von Unit } j \quad (43)$$

$$y_{Si} = y_{Sj} \quad \text{Unit } i \text{ befindet sich auf der südlichen Innenkante von Unit } j \quad (44)$$

$$x_{Ei} = x_{Ej} \quad \text{Unit } i \text{ befindet sich auf der östlichen Innenkante von Unit } j \quad (45)$$

$$x_{Wi} = x_{Wj} \quad \text{Unit } i \text{ befindet sich auf der westlichen Innenkante von Unit } j \quad (46)$$

### Prohibit Connection (PConn)

In einer Grundrissanordnung gibt es eine große Anzahl von Flächen, die in keinerlei Beziehung zueinander stehen und daher nicht explizit mit Bedingungen belegt werden. Fehlende Bedingungen können jedoch dazu führen, dass Flächen nebeneinander angeordnet werden, bei denen eine räumliche Trennung gewahrt werden muss. So sollte beispielsweise das Fluchttreppenhaus eines Gebäudes nicht direkt neben dem Haupttreppenhaus liegen. Das Prohibit Connection Constraint wird immer dann verwendet, wenn Flächen gezielt voneinander getrennt werden müssen. Zur Bestimmung des Mindestabstandes zwischen diesen Flächen können ein Faktor  $\mathcal{E}$  oder Obere und Untere Schranken verwendet werden. Der Mindestabstand muss größer Null sein. Die nachfolgende Definition bewirkt, dass Unit  $i$  in keiner geographischen Himmelsrichtung eine Verbindung zu Unit  $j$  hat:

$$(y_{Ni} < y_{Sj})OR(y_{Nj} < y_{Si})OR(x_{Ei} < x_{Wj})OR(x_{Ej} < x_{Wi}) \quad (47)$$

$$y_{Ni} < y_{Sj} \quad \text{Unit } i \text{ hat keine südliche Verbindung zu Unit } j \quad (48)$$

$$y_{Nj} < y_{Si} \quad \text{Unit } i \text{ hat keine nördliche Verbindung zu Unit } j \quad (49)$$

$$x_{Ei} < x_{Wj} \quad \text{Unit } i \text{ hat keine westliche Verbindung zu Unit } j \quad (50)$$

$$x_{Ej} < x_{wi} \quad \text{Unit } i \text{ hat keine östliche Verbindung zu Unit } j \quad (51)$$

### Design Constraints

Design Constraints übernehmen bei der Anordnung der Units zwei wichtige Aufgaben. Zum einen kann der Architekt durch Festlegung von Relationen zwischen Variablen unmittelbar auf das Erscheinungsbild des Entwurfes Einfluss nehmen, zum anderen können bei Solvern, die nur eine Zielfunktion unterstützen (im ggs. zu Multi-Objective Optimization Goals), diese Zielfunktionen in Form von Constraints definiert werden. Soll beispielsweise eine Unit  $i$  ein bestimmtes Flächenmaß aufweisen, können über Obere und Untere Schranken den Variablen für Länge und Breite der Unit Werte zugewiesen werden, welche die im Constraint beinhaltete Zielfunktion erfüllen. Diese Vorgehensweise ist auch deshalb sinnvoll, da quadratische Zielfunktionen in Optimierungsprogrammen äußerst problematisch sind und entweder gar nicht oder nur in nicht praktikabler Zeit gelöst werden. Es ist ferner möglich, den Erfüllungsgrad der als Constraints definierten Zielfunktionen durch Variablen zu ermitteln und diese mit Hilfe von Design Constraints zu gewichten (vgl. Kap. 5.3.1). Bei unmöglicher Erfüllung aller Constraints kann der Architekt dadurch die Erfüllung bestimmter Constraints forcieren. Design Constraints können als weiche Bedingungen bezeichnet werden, wohingegen die zuvor genannten Constraints harte Bedingungen darstellen. Weiche Bedingungen ermöglichen dem Architekten, gezielt gestalterischen Einfluss auf den Gebäudeentwurf zu nehmen, beispielsweise durch die Spezifikation von Proportionsmaßen zwischen Raumbauteilen oder Raumgruppen. Harte Bedingungen spezifizieren Positionsrelationen zwischen einzelnen Units und gewährleisten die funktionalen Aspekte einer Entwurfslösung. Sinngemäß sind Design Constraints daher frei definierbar und unterliegen den Erfordernissen der jeweiligen Optimierungsaufgabe. Zu den Design Constraints gehören Strecken-, Umfang-, und Flächenspezifikationen, Proportionen (z.B.: quadratische Räume, Längen- / Breitenverhältnisse, Maßzahlen des Goldenen Schnitts) und Symmetrien.

### Length Constraints, Area Constraints, Perimeter Constraints

Die nachfolgenden Constraints spezifizieren eine Strecke, den Flächeninhalt und den Umfang einer Unit  $i$  in Bezug zu einem Parameter  $n$ :

$$\begin{array}{ll}
 l_i & \text{Länge Unit } i \\
 (l_i \leq n) \text{ OR } (l_i = n) \text{ OR } (l_i \geq n) ; i = n & \text{Length Constraint} \quad (52)
 \end{array}$$

$$\begin{array}{ll}
 F_i & \text{Flächeninhalt Unit } i \\
 (F_i \leq n) \text{ OR } (F_i = n) \text{ OR } (F_i \geq n) ; i = n & \text{Area Constraint} \quad (53)
 \end{array}$$

$$\begin{array}{ll}
 U_i & \text{Umfang Unit } i \\
 (U_i \leq n) \text{ OR } (U_i = n) \text{ OR } (U_i \geq n) ; i = n & \text{Perimeter Constraint} \quad (54)
 \end{array}$$

### Aspect Ratio Constraints

Das nachfolgende Constraint definiert quadratische Units einer endlichen Indexmenge  $I_1$  (Flächen des Raumprogramms):

$$\sum_{k \in I_1} (\Delta x_k - \Delta y_k)^2 \rightarrow \min \quad \text{Quadratische Proportionen} \quad (55)$$

### Symmetry Constraints

Das nachfolgende Constraint definiert Symmetrie zwischen zwei Units  $i, j$  in Bezug zu den Mittelachsen einer Unit  $m$ :

$$(FSymm\ i, j - m) \quad \text{Symmetrie von Unit } i, j \text{ zu Unit } m \quad (56)$$

$$b_i, l_i \quad \text{Breite, Länge Unit } i$$

$$(y_{Si} = y_{Sj}) \text{ AND } (l_i = l_j) \text{ AND } (b_i = b_j) \text{ AND} \\ \left( x_{Wm} + \frac{b_m}{2} - x_{Wi} - \frac{b_i}{2} = x_{Wj} + \frac{b_j}{2} - x_{Wm} - \frac{b_m}{2} \right)$$

OR

$$(y_{Wi} = y_{Wj}) \text{ AND } (l_i = l_j) \text{ AND } (b_i = b_j) \text{ AND} \\ \left( y_{Sm} + \frac{l_m}{2} - y_{Si} - \frac{l_i}{2} = y_{Sj} + \frac{l_j}{2} - y_{Sm} - \frac{l_m}{2} \right)$$

### Constraint-Kombinationen

Diese Constraints finden, in Kombination mit einer oder mehreren Void-Units ( $k, l$ ) zur Erzeugung rechtwinkliger Grundrissanordnungen, Verwendung. Das nachfolgend aufgeführte Constraintpaar besteht aus F2Border und FConn Constraints und wird zur Positionierung einer Unit  $i$  an einer der sechs Innenkanten der durch Verwendung der Void-Unit  $k$  rechtwinkligen Unit  $j$  (i.d.R. Grundunit  $R$ ) angewendet. Voraussetzung ist die Existenz der obligatorischen Finside ( $i, j$ ) und Pinter Constraints ( $i, k$ ):

$$(F2Border\ i - j) \text{ OR } (FConn\ i - k) \quad \text{Constraint-Kombination} \quad (57)$$

$$((y_{Ni} = y_{Nj}) \text{ OR } (y_{Si} = y_{Sj}) \text{ OR } (x_{Ei} = x_{Ej}) \text{ OR } (x_{Wi} = x_{Wj}))$$

OR

$$((y_{Nk} = y_{Si}) \text{ OR } (y_{Sk} = y_{Ni}) \text{ AND } \min(x_{Ek} - x_{Wi}, x_{Ei} - x_{Wk}) \geq 0)$$

OR

$$((x_{Ek} = x_{Wi}) \text{ OR } (x_{Wk} = x_{Ei}) \text{ AND } \min(y_{Nk} - y_{Si}, y_{Ni} - y_{Sk}) \geq 0)$$

### 5.3.2.4 Obere und Untere Schranken

Der Einsatz Oberer und Unterer Schranken in Design Constraints kann den Lösungsraum einer Optimierungsaufgabe dramatisch einschränken und führt dadurch zu einer Reduzierung der Berechnungskosten. Die nachfolgenden Constraints spezifizieren eine Strecke, den Flächeninhalt und den Umfang einer Unit  $i$ , als Wert zwischen einer Oberen (U = Upper) und Unteren (L = Lower) Schranke:

$$\begin{array}{ll}
 l_i & \text{Länge Unit } i \\
 l_{iL} \leq l_i \leq l_{iU} ; i = 1, 2, \dots, n & \text{Length Bounds} \quad (58)
 \end{array}$$

$$\begin{array}{ll}
 F_i & \text{Flächeninhalt Unit } i \\
 F_{iL} \leq F_i \leq F_{iU} ; i = 1, 2, \dots, n & \text{Area Bounds} \quad (59)
 \end{array}$$

$$\begin{array}{ll}
 U_i & \text{Umfang Unit } i \\
 U_{iL} \leq U_i \leq U_{iU} ; i = 1, 2, \dots, n & \text{Perimeter Bounds} \quad (60)
 \end{array}$$

Obere und Untere Schranken werden in der Regel als Design Constraints definiert und beinhalten Regeln, die gestalterischen aber auch funktionalen Aspekten unterliegen. So ist die Verwendung einer Unteren Schranke beispielsweise bei der Festlegung der Mindestbreite eines Fluchtweges unabdingbar. Da nicht alle Optimierungsalgorithmen die explizite Angabe Oberer und Unterer Schranken unterstützen, wird hier insbesondere die Möglichkeit der Definition als Constraint hervorgehoben. Programmiersprachen wie OPL (Optimization Programming Language) unterstützen die Deklaration von Variablen mit Angabe von Wertebereichen (vgl. Kap. 7.3). So kann eine Schranke beispielsweise in der Form:

```
var int+ Ni in 0..86;
```

für eine unabhängige, ganzzahlige Design Variable mit einem Wertebereich von 0 bis 86 definiert werden. Abhängige Variablen (Response Variablen) können nicht auf diese Weise deklariert werden. Die Einschränkung muss daher durch eine Constraint-Definition oder eine Zielfunktion erfolgen.

### 5.3.2.5 Zielfunktionen

In der Regel verwenden Optimierungsalgorithmen nur eine Zielfunktion zur Beschreibung des primären Optimierungszieles. Kapitel 5.3.1 zeigt, dass es ebenfalls möglich ist, mehrere Zielfunktionen als Nebenbedingungen zu konstruieren oder auch gänzlich auf die Definition der Zielfunktion zu verzichten (Constraint Satisfaction). In der Revitalisierung spielt die vollständige Erfüllung des Raumprogramms eine übergeordnete Rolle, d.h. die Revitalisierungsmaßnahme wird bezüglich dieses Erfüllungsgrades mit einer potentiellen Neubaumaßnahme verglichen. Das übergeordnete Optimierungsziel besteht daher aus einer Vielzahl untergeordneter Ziele, die

im Wesentlichen bereits durch die Spezifikation der Nebenbedingungen (z.B. Area Constraints) abgedeckt sind. Da in den seltensten Fällen das Flächenangebot des zu revitalisierenden Gebäudes, dem von einem konkreten Gebäude unabhängigen, idealen Flächenprogramm entspricht, können Optimierungsalgorithmen bei exakter Benennung des Flächenbedarfs in den Design Constraints in der Regel keine Lösung finden. Es ist daher sinnvoll, den Flächenbedarf der Units als Teilfunktionen durch Obere und Untere Schranken für Design Variablen (Länge, Breite) sowie als Constraints für Response Variablen (Flächen oder Umfang) zu deklarieren und gleichzeitig eine zu minimierende oder maximierende Zielfunktion als Summe der Teilfunktionen aufzustellen. Auf diese Weise wird dem Algorithmus ein Handlungsspielraum eingeräumt, der ihm ermöglicht, das Ergebnis an die vollständige Erfüllung des Raumprogramms anzunähern. Teilfunktionen, deren Erfüllungsgrad von besonderer Bedeutung ist, können durch die Vergabe von Gewichten in ihrer Erfüllung begünstigt werden (vgl. Formel (4)).

Die nachfolgende Zielfunktion maximiert die Summe der Flächeninhalte einer endlichen Indexmenge  $I_2$  (Flächen des Raumprogramms):

$$\sum_{k \in I_2} \Delta x_k \Delta y_k \rightarrow \max \quad \text{Zielfunktion Flächenmaximierung} \quad (61)$$

Da die Verwendung quadratischer Zielfunktionen, die aus der Deklaration einer Zielfunktion zur Bestimmung des Flächeninhaltes einer Indexmenge resultiert, nicht von allen Optimierungsalgorithmen unterstützt wird, kann die folgende Zielfunktion als Approximation der Flächeninhalte verwendet werden. Sie maximiert die Summe der Flächenumfänge einer endlichen Indexmenge  $I_3$  (Flächen des Raumprogramms):

$$\sum_{k \in I_3} \Delta x_k + \Delta y_k \rightarrow \max \quad \text{Zielfunktion Umfangmaximierung} \quad (62)$$

### 5.3.2.6 Suchstrategien

Viele kombinatorische Optimierungsprobleme sind NP-complete, d.h. sie sind mit nichtdeterministisch polynomialem Aufwand nicht zu lösen. Es wird allgemein angenommen, dass kein genereller Algorithmus existiert oder je existieren wird, welcher in der Lage ist, diese Probleme zu lösen [Gritzmann und Brandenburg, 2003]. Um derartige Optimierungsprobleme dennoch näherungsweise lösen zu können, bieten Optimierungssprachen wie OPL die Möglichkeit, Suchstrategien zu definieren. Herkömmliche Modellierungssprachen oder mathematische Programmiersprachen erlauben keinen Zugriff auf deren Suchmechanismen. Die Definition einer Suchstrategie kann die Berechnungszeit eines Optimierungsproblems fundamental verringern. Dies ist insbesondere bei Maximierungs- und Minimierungsproblemen der Fall. Bei Constraint Satisfaction Problemen wird unter Verwendung von OPL gemeinhin auch ohne Suchstrategie relativ schnell eine Lösung gefunden. Obere und Untere Schranken stellen in gewisser Weise bereits eine Suchstrategie dar, da sie den Wertebereich einer Variablen einschränken. Die explizite Vorgabe einer Suchstrategie stellt keine ideologische Beschränkung des maschinellen Optimierungsvorgangs dar. Es wird allerdings offensichtlich, dass menschliche Problemlösungs-



strategien den numerischen Strategien in mancherlei Hinsicht noch überlegen sind. Die nachfolgende OPL-Definition weist einer Variablen  $\Delta x_i$  (Breite Unit  $i$ ) Werte von 0 bis 25 in absteigender Reihenfolge zu:

```
// SEARCH PROCEDURES
search {
tryall (i in 0..25 ordered by decreasing i)
   $\Delta x_i = i$ ;
};
```

Suchstrategien können ebenfalls verwendet werden, um die Wertebereiche mehrerer gleichartiger Variablen miteinander zu koordinieren. Klare Grundrisslösungen zeichnen sich in der Regel durch kongruente Längen- und Breitenverhältnisse ähnlicher Units aus. Die nachfolgende Prozedur weist der Breite mehrerer Units Variablenwerte zwischen 0 und 10 und der Länge Variablenwerte zwischen 0 und 12, mit einer Iteration von 2 zu. Auf diese Weise kann die Ausrichtung von Units beispielsweise an existente Stützenraster gebunden werden:

```
// SEARCH PROCEDURES
search {
tryall (i in 0..10 : i mod 2 = 0)
   $\Delta x_i = i \ \& \ \Delta x_j = i \ \& \ \Delta x_k = i \ \& \ \Delta x_l = i \ \& \ \Delta x_m = i$ ;
tryall (j in 0..12 : j mod 2 = 0)
   $\Delta y_i = j \ \& \ \Delta y_j = j \ \& \ \Delta y_k = j \ \& \ \Delta y_l = j \ \& \ \Delta y_m = j$ ;
};
```

### 5.3.3 Zerstörungsarmes Modell



Die Einschränkungen des Zerstörungsreichen Modells auf rechtwinklige Ausprägungen der Units werden in diesem Modell aufgehoben. Der Vorteil des Zerstörungsarmen Modells ist, dass es auf jede Grundrisskonstellation angewendet werden kann. Insbesondere komplexe Bestandsbauten ohne ordnende Strukturen können mit diesem Modell beplant werden. Der Zweck dieses Modells ist es, wie in Kapitel 3 erläutert, den Tausch von Flächen (Units) zu bewerkstelligen, ohne bauliche Eingriffe vorzunehmen. Unter Tauschen sei verstanden, dass eine Unit  $i$  mit einer Unit  $j$  auf eine Weise getauscht wird, dass die Namen (respektive die Nutzungen) getauscht werden. Dieses Tauschen geschieht durch die Zuordnung einer Unit zu einer dem Raumprogramm entlehnten Gruppe und wird durch Veränderung des Namens (*label*) angezeigt. Die Units bleiben geometrisch unverändert, aus Unit  $i$  wird Unit  $j$  und umgekehrt. Dieses Tauschen findet allerdings nicht nur zwischen zwei Units, sondern zwischen allen Units eines Grundrisses statt. Als Eingabeparameter dienen dazu die Angaben des Raumprogramms. Diese beinhalten Angaben über Gruppen, d.h. zusammengehörige, benachbarte Units, sowie über deren Größe und Anzahl.

Das Zerstörungssarme Modell arbeitet gänzlich ohne geometrisch basierte Nebenbedingungen zwischen den Parametern einzelner Units. Es wird lediglich die Benachbarung zwischen existenten Units definiert.

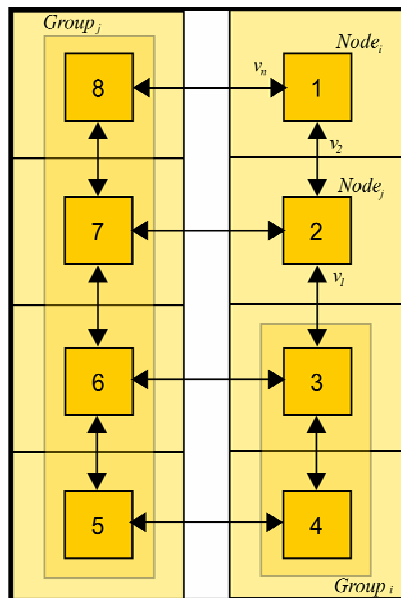


Abb. 5.9 Repräsentation eines Grundrisses durch einen Graphen

### 5.3.3.1 Konstanten

Das Modell besitzt keine konstanten Parameter, die im Optimierungsprozess eingesetzt werden müssen. Das Wesen dieses Modells ist es, die Position der Units konstant zu halten. Zur Erweiterung des potentiellen Lösungsraumes können die Units jedoch in ihrer Größe und Benachbarung durch Veränderung des Graphen variiert werden.

### 5.3.3.2 Variablen

Die wichtigste Variable im Optimierungsprozess des ZA-Modells wird aus einer Adjazenzmatrix gebildet. Diese Matrix beschreibt die Benachbarung zwischen den Units des existenten Grundrisses. Die Art, wie diese Benachbarung definiert wird, ist variabel und kann vom Architekten gemäß den spezifischen Anforderungen des Projektes definiert werden. Unter Benachbarung muss nicht die direkte Nachbarschaft zu einer anderen Unit verstanden werden. Auch sich an einer Verkehrsfläche gegenüberliegende Units können durchaus als Nachbarn gelten. Jede Unit bildet innerhalb des Graphen einen Knoten. Ziel ist es, innerhalb aller Knoten des Graphen, die aus mehreren Units bestehenden Gruppen des Raumprogramms zu finden. Existiert keine Lösung, ist es möglich die Nebenbedingungen in Form der nachfolgend genannten Variablen zu

verändern. Diese Veränderung entspricht einer Anpassung des Raumprogramms. Unter dem Aspekt der Nachhaltigkeit ist dieses die zu bevorzugende Vorgehensweise:

- nbreOfGroups* – Anzahl der gesuchten Gruppen zusammengehöriger Units des Raumprogramms
- sizeGroups* – Größe der Gruppen zusammengehöriger Units des Raumprogramms in Quadratmeter
- groups* – Anzahl der Units innerhalb einer Gruppe

Eine weitere, jedoch weniger ökologische Möglichkeit zur Lösungsfindung, ist die Veränderung der Adjazenz-Matrix. Durch zusätzliche strukturelle oder organisatorische Zerlegung des Grundrisses in kleinere Einheiten wird die Matrix vergrößert, so dass neue Raumkonstellationen entstehen. Diese Vorgehensweise bedingt eine Änderung der nachfolgend genannten Variablen:

- A* – Benachbarung der existenten Units des Grundrisses
- nbreOfNodes* – Anzahl der existenten Units des Grundrisses
- sizeNodes* – Größe der existenten Units des Grundrisses in Quadratmeter

Eine Erweiterung der Adjazenz-Matrix in Form der Benachbarung resultiert noch nicht zwingend in einer baulichen Veränderung des Grundrisses. Diese sehr ökologische Vorgehensweise bedingt jedoch unter Umständen größere Entfernungen zwischen benachbarten Knoten. Wird allerdings die Anzahl und somit auch die Größe der Units verändert, sind damit in der Praxis bauliche Veränderungen des Gebäudes verbunden. In vielen Fällen stellt dies ökologisch allerdings ein weniger großes Problem dar, als der klassische Umbau, da die Eingriffe überschaubar sind, wenn die Matrix schrittweise verändert wird. Der Abbruch von Trennwänden oder auch der Einbau neuer Wände führt zur Veränderung der Adjazenz und potentiell zu größerem Spielraum für den Prozess des Tauschens von Flächen.

Als Ergebnis des Optimierungsprozesses werden den Variablen neue Werte zugewiesen. Diese Werte entsprechen der Zugehörigkeit zu einer Gruppe, deren Name (in Abhängigkeit von der Anzahl der Gruppen) der Variable *label* zugewiesen wird:

- label* – Namen, der aus dem Raumprogramm gebildeten Gruppen

### 5.3.3.3 Nebenbedingungen

Das Zerstörungsarme Modell baut auf der Benachbarung der existenten Units und der Benachbarung von Units innerhalb der gefundenen Gruppen auf. Benachbarung ist gegeben, wenn jede Unit mindestens eine weitere Unit als Nachbar besitzt. Ferner müssen folgende Nebenbedingungen für eine plausible Lösung erfüllt sein:

- Die Anzahl der Units einer Gruppe stimmt mit der definierten Anzahl der Units in der Gruppe überein.
- Die Summe der Werte aller Units einer Gruppe (in Quadratmeter) ist größer oder gleich der definierten Gruppengröße.
- Die Units einer Gruppe müssen zusammenhängend sein.

### 5.3.3.4 Obere und Untere Schranken

Das Zerstörungsarme Modell benutzt Obere und Untere Schranken für die Bestimmung der prozentualen Abweichung von der gem. Raumprogramm geforderten Gesamtgruppengröße. Anders als beim ZR-Modell werden die Schranken hier nicht zur Einschränkung, sondern zur Erweiterung des potentiellen Lösungsraumes benutzt.

- |            |   |
|------------|---|
| <i>pso</i> | – Obere Schranke, d.h. Prozentsatz, welcher von der geforderten Gruppengröße maximal erfüllt sein darf (i.A. $\geq 100\%$ )     |
| <i>psu</i> | – Untere Schranke, d.h. Prozentsatz, welcher von der geforderten Gruppengröße mindestens erfüllt sein muss (i.A. $\leq 100\%$ ) |

### 5.3.3.5 Zielfunktionen

In diesem Modell sind Zielfunktionen theoretisch obsolet, da die Erfüllung der Nebenbedingungen die Erfüllung des Zieles, d.h. das Auffinden der definierten Gruppen, der entsprechenden Gruppengröße und der jeweiligen Anzahl der Units innerhalb der Gruppen beinhaltet (Constraint Satisfaction). Es ist jedoch möglich, übergeordnete Ziele zu definieren (vgl. Kap. 5.3.1). Diese übergeordneten Ziele gewichten auch hier die Zielerfüllung. Werden innerhalb eines Grundrisses beispielsweise alle Bedingungen mit einer Ausnahme erfüllt (z.B. die Größe einer gefundenen Gruppe ist 5 qm zu klein oder zu groß), wäre es unsinnig, in der Praxis nicht dennoch den Flächentausch vorzunehmen. Aus diesem Grund finden derartige Lösungen, welche durch die Verwendung Oberer und Unterer Schranken ermöglicht werden, in Form gewichteter Ziele Berücksichtigung im Performancemodell (vgl. Kap. 7.3.4.1, Beispiel 4).

### 5.3.3.6 Suchstrategien

Das Zerstörungssarme Modell benutzt keine Suchstrategien, um den Lösungsraum des Optimierungsproblems einzuschränken. Es wird lediglich eine Breitensuche [Jungnickel, 2005] verwendet, um den Zusammenhang des Graphen sicher zu stellen.

## 5.4 Performancemodell

Das Performancemodell dient der Messung der Performance der Optimierungsergebnisse. Wesentliche Bestandteile dieses Modells sind die graphische Evaluierung und Gewichtung der verwendeten Eingabedaten aus dem Optimierungsmodell, sowie die Messung des Erfüllungsgrades der Constraints. Das Performancemodell muss die graphische Anzeige von Varianten mit Angabe der jeweiligen Gesamtpformance sicherstellen. Ferner ist die Veränderung der Ausgabeparameter als erneuter Input für weitere Optimierungsrechnungen erforderlich. Insbesondere die Ergebnisse von Optimierungsrechnungen mit zu minimierenden und zu maximierenden Zielfunktionen erfordern die Überführung der ermittelten Ergebnisse in Constraint Satisfaction Probleme, um bei weiteren Optimierungsrechnungen eine größere Anzahl von Lösungen gleicher Güte zu erhalten.

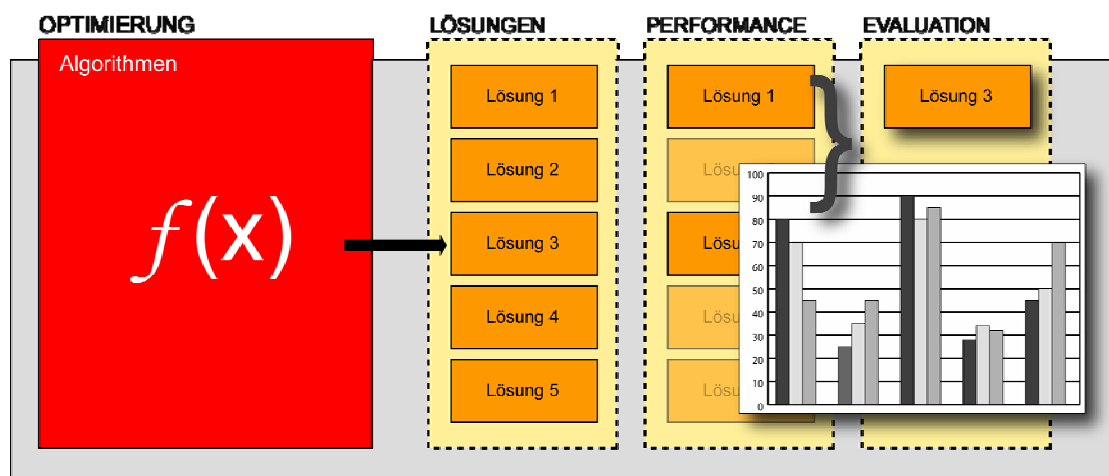


Abb. 5.10 Schematische Darstellung des Performancemodells

Die in der Systemkonzeption beschriebenen Geometrie- und Optimierungsmodelle des Zerstörungsreichen und des Zerstörungssarmen Modells werden im folgenden Kapitel in mathematische Modelle überführt.

## **6 Mathematische Modellbildung**

## 6.1 Mathematisches Modell 1 (Zerstörungsreiches Modell)

Dieses Modell basiert auf den in Kapitel 5.3.2 beschriebenen Grundlagen. Es kann unabhängig von spezifischen Optimierungsverfahren oder Programmiersprachen verwendet werden.

### 6.1.1 Ausgangsmodell

<p>Geometriemodell</p>		<p>Nr.</p>
<p>Konventionen</p>	<p>Unit i, j</p> <p>Unit R</p>	<p>Bezeichnung der Grundflächen neuer Nutzungseinheiten</p> <p>Bezeichnung der Grundfläche des Gebäudebestandes</p>

	$(I_1, I_2, I_3)$	Endliche Indexmengen, d.h. die ausgewählten Units	
Konstanten	$(x_R, y_R) := (0, 0)$ $(N_R, S_R, E_R, W_R)$ $(y_{NR}, y_{SR}, x_{ER}, x_{WR})$ $(b_R, l_R)$	Referenzpunkt der Unit R Kantenabstände vom Referenzpunkt der Unit R Kantenpositionen der Unit R Breite, Länge der Unit R	
Variablen			
Design Variablen	$(x_i, y_i) ; (x_j, y_j)$ $(N_i, S_i, E_i, W_i); (N_j, S_j, E_j, W_j)$	Referenzpunkte der Units i; j Kantenabstände von den Referenzpunkten der Units i; j	
Response Variablen	$(y_{Ni}, y_{Si}, x_{Ei}, x_{Wi}); (y_{Nj}, y_{Sj}, x_{Ej}, x_{Wj})$ $(b_i, l_i) ; (b_j, l_j)$ $F_i; F_j$	Kantenpositionen der Units i; j $y_{Ni} = y_i + N_i$ $y_{Si} = y_i - S_i$ $x_{Ei} = x_i + E_i$ $x_{Wi} = x_i - W_i$ Breite, Länge der Units i; j $b_i = W_i + E_i ; l_i = N_i + S_i$ $b_j = W_j + E_j ; l_j = N_j + S_j$ Flächeninhalt der Units i; j $F_i = b_i \times l_i ; F_j = b_j \times l_j$	



Nebenbedingungen		
Non-Negative	$x_i, y_i \in \{\mathbb{R}_+\}$ Nichtnegativitätsbedingung $(N_i, S_i, E_i, W_i) \in \{\mathbb{R}_+\}$ Nichtnegativitätsbedingung	(63)
Force Inside	$(y_{Ni} \leq y_{Nj}) \text{ AND } (y_{Si} \leq y_{Si}) \text{ AND } (x_{Ei} \leq x_{Ej}) \text{ AND } (x_{Wj} \leq x_{Wi})$ $y_{Ni} \leq y_{Nj}$ Unit i südl. der nördlichen Innenkante von Unit j $y_{Si} \leq y_{Si}$ Unit i nördl. der südlichen Innenante von Unit j $x_{Ei} \leq x_{Ej}$ Unit i westl. der östlichen Innenkante von Unit j $x_{Wj} \leq x_{Wi}$ Unit i östl. der westlichen Innenkante von Unit j  Die nachfolgende MAX-Funktion bewirkt, dass Unit i vollständig innerhalb Unit j liegt:	
	$\max(y_{Ni} - y_{Nj}, y_{Si} - y_{Si}, x_{Ei} - x_{Ej}, x_{Wj} - x_{Wi}) \leq 0$	(64)
Prohibit Intersection	$(y_{Ni} \leq y_{Sj}) \text{ OR } (y_{Nj} \leq y_{Si}) \text{ OR } (x_{Ei} \leq x_{Wj}) \text{ OR } (x_{Ej} \leq x_{Wi})$ $y_{Ni} \leq y_{Sj}$ Unit i befindet sich südlich von Unit j $y_{Nj} \leq y_{Si}$ Unit i befindet sich nördlich von Unit j $x_{Ei} \leq x_{Wj}$ Unit i befindet sich westlich von Unit j $x_{Ej} \leq x_{Wi}$ Unit i befindet sich östlich von Unit j  Die nachfolgende MIN-Funktion bewirkt, dass es in keiner geographischen Himmelsrichtung eine Überschneidung zwischen Unit i und Unit j gibt:	
	$\min(y_{Ni} - y_{Sj}, y_{Nj} - y_{Si}, x_{Ei} - x_{Wj}, x_{Ej} - x_{Wi}) \leq 0$	(65)

<p>Force Connection</p>	<p> <math>(y_{Nj} = y_{Si}) OR (y_{Sj} = y_{Ni}) AND \min(x_{Ej} - x_{Wi}, x_{Ei} - x_{Wj}) \geq 0</math>  <i>OR</i>  <math>(x_{Ej} = x_{Wi}) OR (x_{Wj} = x_{Ei}) AND \min(y_{Nj} - y_{Si}, y_{Ni} - y_{Sj}) \geq 0</math> </p> <p>Unit i befindet sich nördlich und berührt Unit j:</p> <p><math>FConnN1(i, j) : y_{Nj} - y_{Si} = 0</math></p> <p><math>FConnN2(i, j) : \min(x_{Ej} - x_{Wi}, x_{Ei} - x_{Wj}) \geq 0</math></p> <p>Unit i befindet sich südlich und berührt Unit j:</p> <p><math>FConnS1(i, j) : y_{Sj} - y_{Ni} = 0</math></p> <p><math>FConnS2(i, j) : \min(x_{Ej} - x_{Wi}, x_{Ei} - x_{Wj}) \geq 0</math></p> <p>Unit i befindet sich östlich und berührt Unit j:</p> <p><math>FConnE1(i, j) : x_{Ej} - x_{Wi} = 0</math></p> <p><math>FConnE2(i, j) : \min(y_{Nj} - y_{Si}, y_{Ni} - y_{Sj}) \geq 0</math></p> <p>Unit i befindet sich westlich und berührt Unit j:</p> <p><math>FConnW1(i, j) : x_{Wj} - x_{Ei} = 0</math></p> <p><math>FConnW2(i, j) : \min(y_{Nj} - y_{Si}, y_{Ni} - y_{Sj}) \geq 0</math></p> <p>Die nachfolgende MIN-Funktion bewirkt, dass Unit i in einer beliebigen geographischen Himmelsrichtung eine Verbindung zu Unit j hat:</p>	
	$\min \left\{ \begin{array}{l} \max \{ x_{Wj} - x_{Ei}, x_{Wi} - x_{Ej}, y_{Ni} - y_{Sj}, y_{Sj} - y_{Ni} \}, \\ \max \{ x_{Wj} - x_{Ei}, x_{Wi} - x_{Ej}, y_{Nj} - y_{Si}, y_{Si} - y_{Nj} \}, \\ \max \{ y_{Sj} - y_{Ni}, y_{Si} - y_{Nj}, x_{Ei} - x_{Wj}, x_{Wj} - x_{Ei} \}, \\ \max \{ y_{Sj} - y_{Ni}, y_{Si} - y_{Nj}, x_{Ej} - x_{Wi}, x_{Wi} - x_{Ej} \} \end{array} \right\} \leq 0$	(66)

<p>Force Outside</p>	<p> <math>(y_{Ni} \leq y_{SR})OR(y_{NR} \leq y_{Si})OR(x_{Ei} \leq x_{WR})OR(x_{ER} \leq x_{Wi})</math>   <math>y_{Ni} - y_{SR} \leq 0</math>      Unit i südlich außerhalb oder auf südlicher Außenkante von R   <math>y_{NR} - y_{Si} \leq 0</math>      Unit i nördlich außerhalb oder auf nördlicher Außenkante von R   <math>x_{Ei} - x_{WR} \leq 0</math>      Unit i westlich außerhalb oder auf westlicher Außenkante von R   <math>x_{ER} - x_{Wi} \leq 0</math>      Unit i östlich außerhalb oder auf östlicher Außenkante von R   Die nachfolgende MIN-Funktion bewirkt, dass Unit i in einer beliebigen geographischen Himmelsrichtung außerhalb von Unit R liegt: </p>	
	<p><math>\min(y_{Ni} - y_{SR}, y_{NR} - y_{Si}, x_{Ei} - x_{WR}, x_{ER} - x_{Wi}) \leq 0</math></p>	<p>(67)</p>
<p>Force To Border</p>	<p> <math>(y_{Ni} = y_{Nj})OR(y_{Si} = y_{Sj})OR(x_{Ei} = x_{Ej})OR(x_{Wi} = x_{Wj})</math>   <math>y_{Ni} - y_{Nj} = 0</math>      Unit i auf der nördlichen Innenkante von j   <math>y_{Si} - y_{Sj} = 0</math>      Unit i auf der südlichen Innenkante von j   <math>x_{Ei} - x_{Ej} = 0</math>      Unit i auf der östlichen Innenkante von j   <math>x_{Wi} - x_{Wj} = 0</math>      Unit i auf der westlichen Innenkante von j   Die nachfolgende MIN-Funktion bewirkt, dass Unit i in einer beliebigen geographischen Himmelsrichtung auf einer Innenkante von Unit j liegt: </p>	
	<p><math>\min(y_{Ni} - y_{Nj}, y_{Si} - y_{Sj}, x_{Ei} - x_{Ej}, x_{Wi} - x_{Wj}) = 0</math></p>	<p>(68)</p>

<p>Prohibit Connection</p>	$(y_{Ni} < y_{Sj}) OR (y_{Nj} < y_{Si}) OR (x_{Ei} < x_{Wj}) OR (x_{Ej} < x_{Wi})$ <p> <math>y_{Ni} &lt; y_{Sj}</math>    Unit i hat keine südliche Verbindung zu Unit j  <math>y_{Nj} &lt; y_{Si}</math>    Unit i hat keine nördliche Verbindung zu Unit j  <math>x_{Ei} &lt; x_{Wj}</math>    Unit i hat keine westliche Verbindung zu Unit j  <math>x_{Ej} &lt; x_{Wi}</math>    Unit i hat keine östliche Verbindung zu Unit j         </p> <p>Die nachfolgende MIN-Funktion bewirkt, dass Unit i in keiner geographischen Himmelsrichtung eine Verbindung zu Unit j hat:</p>	
	$\min(y_{Ni} - y_{Sj}, y_{Nj} - y_{Si}, x_{Ei} - x_{Wj}, x_{Ej} - x_{Wi}) < 0$	(69)
<p>Design Constraints</p>	<p>Length Constraint:</p> $l_i \begin{cases} \leq z_i, & \text{falls Länge von Unit i nach oben begrenzt sein soll} \\ = z_i, & \text{falls Länge von Unit i genau bestimmt sein soll} \\ \geq z_i, & \text{falls Länge von Unit i nach unten begrenzt sein soll} \end{cases}$ <p>Area Constraint:</p> $F_i \begin{cases} \leq z_i, & \text{falls Flächeninhalt von Unit i nach oben begrenzt sein soll} \\ = z_i, & \text{falls Flächeninhalt von Unit i genau bestimmt sein soll} \\ \geq z_i, & \text{falls Flächeninhalt von Unit i nach unten begrenzt sein soll} \end{cases}$ <p>Perimeter Constraint:</p> $U_i \begin{cases} \leq z_i, & \text{falls Umfang von Unit i nach oben begrenzt sein soll} \\ = z_i, & \text{falls Umfang von Unit i genau bestimmt sein soll} \\ \geq z_i, & \text{falls Umfang von Unit i nach unten begrenzt sein soll} \end{cases}$ <p><math>(z_i \in \mathbb{R}_+ \text{ fest, } i \in \{1, \dots, n\})</math></p>	

Zielfunktionen		
Flächen- Maximierung	$\sum_{k \in I_1} b_k l_k \rightarrow \max$	
Umfang- Maximierung	$\sum_{k \in I_2} b_k + l_k \rightarrow \max$	
Annäherung Raum- fläche an Zielgröße	$F_0 > 0$ geg. ; $(b_k l_k - F_0)^2 \rightarrow \min$	
Erzeugung quadratischer Units	$\sum_{k \in I_3} (b_k - l_k)^2 \rightarrow \min$	

6.1.2 Erweiterung (1) des Ausgangsmodells

<p>Geometriemodell</p>			<p>Nr.</p>
<p>Konventionen</p>	<p>Unit i, j  Unit R  (<math>I_1, I_2, I_3</math>)  <math>\varepsilon</math></p>	<p>Bezeichnung der Grundflächen neuer Nutzungseinheiten  Bezeichnung der Grundfläche des Gebäudebestandes  endliche Indextmengen, d.h. die ausgewählten Units  Abstandsstrecke zweier Units i, j</p>	
<p>Konstanten</p>	<p><math>(x_R, y_R) := (0, 0)</math></p>	<p>Referenzpunkt der Unit R</p>	

	$X; Y$	Breite; Länge der Unit R	
Variablen			
Design Variablen	$(x_i, y_i); (x_j, y_j)$	Referenzpunkte der Units i; j	
	$(\Delta x_i, \Delta y_i); (\Delta x_j, \Delta y_j)$	Breite, Länge der Units i; j	
Binäre Variablen	$A_{ij}^X := \left\{ \begin{array}{l} 1, \text{ gdw. Unit i links von Unit j} \\ 0, \text{ sonst} \end{array} \right\}$ $A_{ji}^X := \left\{ \begin{array}{l} 1, \text{ gdw. Unit j links von Unit i} \\ 0, \text{ sonst} \end{array} \right\}$ $A_{ij}^Y := \left\{ \begin{array}{l} 1, \text{ gdw. Unit i unterhalb von Unit j} \\ 0, \text{ sonst} \end{array} \right\}$ $A_{ji}^Y := \left\{ \begin{array}{l} 1, \text{ gdw. Unit j unterhalb von Unit i} \\ 0, \text{ sonst} \end{array} \right\}$ <p>im Fall F2Border:</p> $A_{ij}^X := \left\{ \begin{array}{l} 1, \text{ gdw. Unit i liegt "Ostseite"} \\ 0, \text{ sonst} \end{array} \right\}$ $A_{ji}^X := \left\{ \begin{array}{l} 1, \text{ gdw. Unit i liegt "Westseite"} \\ 0, \text{ sonst} \end{array} \right\}$ $A_{ij}^Y := \left\{ \begin{array}{l} 1, \text{ gdw. Unit i liegt "Nordseite"} \\ 0, \text{ sonst} \end{array} \right\}$ $A_{ji}^Y := \left\{ \begin{array}{l} 1, \text{ gdw. Unit i liegt "Südseite"} \\ 0, \text{ sonst} \end{array} \right\}$		
Nebenbedingungen			
Non-Negative	$x_i, y_i \in \{\mathbb{R}_+\}$	Nichtnegativitätsbedingung	(70)
	$\Delta x_i, \Delta y_i \in \{\mathbb{R}_+\}$	Nichtnegativitätsbedingung	

Force Inside	Die nachfolgenden Funktionen bewirken, dass Unit i vollständig innerhalb Unit j liegt:	
	$x_i \geq x_j$ $y_i \geq y_j$ $x_i + \Delta x_i - (x_j + \Delta x_j) \leq 0$ $y_i + \Delta y_i - (y_j + \Delta y_j) \leq 0$	(71)
Prohibit Intersection	Die nachfolgenden Funktionen bewirken, dass es in keiner geographischen Himmelsrichtung eine Überschneidung zwischen Unit i und Unit j gibt:	
	$x_i + \Delta x_i - X(1 - A_{ij}^X) \leq x_j$ $y_i + \Delta y_i - Y(1 - A_{ij}^Y) \leq y_j$ $A_{ij}^X + A_{ji}^X + A_{ij}^Y + A_{ji}^Y \geq 1$	(72)
Force Connection	Die nachfolgenden Funktionen bewirken, dass Unit i in einer beliebigen geographischen Himmelsrichtung eine Verbindung zu Unit j hat:	
	$y_i + \Delta y_i - Y(1 - A_{ij}^Y) \leq y_j$ $y_i + \Delta y_i + Y(1 - A_{ij}^Y) \geq y_j$ $x_i + \Delta x_i + X(1 - A_{ij}^X) \geq x_j$ $x_j + \Delta x_j + X(1 - A_{ij}^X) \geq x_i$ $x_i + \Delta x_i - X(1 - A_{ij}^X) \leq x_j$ $x_i + \Delta x_i + X(1 - A_{ij}^X) \geq x_j$ $y_i + \Delta y_i + Y(1 - A_{ij}^Y) \geq y_j$ $y_j + \Delta y_j + Y(1 - A_{ij}^Y) \geq y_i$ $A_{ij}^X + A_{ji}^X + A_{ij}^Y + A_{ji}^Y \geq 1$	(73)



Force Outside	Die nachfolgenden Funktionen bewirken, dass Unit i in einer beliebigen geographischen Himmelsrichtung außerhalb von Unit j liegt:	
	$x_i + \Delta x_i - (X + \varepsilon)(1 - A_{ij}^X) \leq x_j - \varepsilon$ $x_j + \Delta x_j - (X + \varepsilon)(1 - A_{ji}^X) \leq x_i - \varepsilon$ $y_i + \Delta y_i - (Y + \varepsilon)(1 - A_{ij}^Y) \leq y_j - \varepsilon$ $y_j + \Delta y_j - (Y + \varepsilon)(1 - A_{ji}^Y) \leq y_i - \varepsilon$ $\varepsilon \geq 0$ $A_{ij}^X + A_{ji}^X + A_{ij}^Y + A_{ji}^Y \geq 1$	
Force To Border	Die nachfolgenden Funktionen bewirken, dass Unit i in einer beliebigen geographischen Himmelsrichtung auf einer Innenkante von Unit j liegt:	
	$x_i + \Delta x_i + X(1 - A_{ij}^X) \geq X$ $x_i - X(1 - A_{ji}^X) \leq 0$ $y_i + \Delta y_i + Y(1 - A_{ij}^Y) \geq Y$ $y_i - Y(1 - A_{ji}^Y) \leq 0$ $A_{ij}^X + A_{ji}^X + A_{ij}^Y + A_{ji}^Y \geq 1$	(74)
Prohibit Connection	Die nachfolgenden Funktionen bewirken, dass Unit i in keiner geographischen Himmelsrichtung eine Verbindung zu Unit j hat:	
	$x_i + \Delta x_i - (X + \varepsilon)(1 - A_{ij}^X) \leq x_j - \varepsilon$ $x_j + \Delta x_j - (X + \varepsilon)(1 - A_{ji}^X) \leq x_i - \varepsilon$ $y_i + \Delta y_i - (Y + \varepsilon)(1 - A_{ij}^Y) \leq y_j - \varepsilon$ $y_j + \Delta y_j - (Y + \varepsilon)(1 - A_{ji}^Y) \leq y_i - \varepsilon$ $\varepsilon > 0$ $A_{ij}^X + A_{ji}^X + A_{ij}^Y + A_{ji}^Y \geq 1$	(75)

<p>Design Constraints</p>	<p>Length Constraint,</p> $\Delta y_i \left\{ \begin{array}{l} \leq z_i, \text{ falls Länge von Unit } i \text{ nach oben begrenzt sein soll} \\ = z_i, \text{ falls Länge von Unit } i \text{ genau bestimmt sein soll} \\ \geq z_i, \text{ falls Länge von Unit } i \text{ nach unten begrenzt sein soll} \end{array} \right\}$ <p>Area Constraint:</p> $F_i \left\{ \begin{array}{l} \leq z_i, \text{ falls Flächeninhalt von Unit } i \text{ nach oben begrenzt sein soll} \\ = z_i, \text{ falls Flächeninhalt von Unit } i \text{ genau bestimmt sein soll} \\ \geq z_i, \text{ falls Flächeninhalt von Unit } i \text{ nach unten begrenzt sein soll} \end{array} \right\}$ <p>Perimeter Constraint:</p> $U_i \left\{ \begin{array}{l} \leq z_i, \text{ falls Umfang von Unit } i \text{ nach oben begrenzt sein soll} \\ = z_i, \text{ falls Umfang von Unit } i \text{ genau bestimmt sein soll} \\ \geq z_i, \text{ falls Umfang von Unit } i \text{ nach unten begrenzt sein soll} \end{array} \right\}$ <p><math>(z_i \in \mathbb{R}_+ \text{ fest, } i \in \{1, \dots, n\})</math></p>	
<p>Zielfunktionen</p>		
<p>Flächen- Maximierung</p>	$\sum_{k \in I_1} \Delta x_k \Delta y_k \rightarrow \max$	
<p>Umfang- Maximierung</p>	$\sum_{k \in I_2} \Delta x_k + \Delta y_k \rightarrow \max$	
<p>Annäherung Raum- fläche an Zielgröße</p>	$F_0 > 0 \text{ geg. ; } (\Delta x_k \Delta y_k - F_0)^2 \rightarrow \min$	
<p>Erzeugung quadratischer Units</p>	$\sum_{k \in I_3} (\Delta x_k - \Delta y_k)^2 \rightarrow \min$	

6.1.3 Erweiterung (2) des Ausgangsmodells

<p>Geometriemodell</p>	<p>The diagram illustrates a geometric model of a building layout. It features a large outer rectangle labeled <math>Unit_{R'}</math> with dimensions <math>X'</math> (width) and <math>Y'</math> (height). Inside this rectangle is a smaller rectangle labeled <math>Unit_R</math> with dimensions <math>X</math> (width) and <math>Y</math> (height). Within <math>Unit_R</math>, there are two smaller rectangles: <math>Unit_i</math> at the bottom-left and <math>Unit_j</math> at the top-right. The dimensions of <math>Unit_i</math> are <math>\Delta x_i</math> and <math>\Delta y_i</math>, and its bottom-left corner is marked with a red dot at point <math>P(x_i, y_i)</math>. The dimensions of <math>Unit_j</math> are <math>\Delta x_j</math> and <math>\Delta y_j</math>, and its bottom-left corner is marked with a red dot at point <math>P(x_j, y_j)</math>. Above <math>Unit_j</math> is a yellow rectangle labeled <math>Unit_m</math>. The bottom-left corner of <math>Unit_R</math> is marked with a red dot at point <math>P(x_R, y_R)</math>, and the bottom-left corner of <math>Unit_{R'}</math> is marked with a red dot at point <math>P(x_{R'}, y_{R'})</math>. The diagram is oriented with North at the top, South at the bottom, West on the left, and East on the right.</p>	<p>Nr.</p>
<p>Konventionen</p>	<p>Unit i, j                      Bezeichnung der Grundflächen neuer Nutzungseinheiten</p> <p>Unit R                              Bezeichnung der Grundfläche des Gebäudebestandes</p> <p>Unit R'                              Bezeichnung der Grundfläche des Grundstücks</p> <p><math>(I_1, I_2, I_3)</math>                      endliche Indexmengen, d.h. die</p>	

		ausgewählten Units	
	$\varepsilon$	Abstandsstrecke zweier Units i, j	
	$\delta$	Berührungsstrecke zweier Units i, j	
Konstanten	$(x_R, y_R)$	Referenzpunkt der Unit R	
	$X; Y$	Breite; Länge der Unit R	
	$(x_{R'}, y_{R'}) := (0, 0)$	Referenzpunkt der Unit R'	
	$X'; Y'$	Breite; Länge der Unit R'	
Variablen			
Design Variablen	$(x_i, y_i); (x_j, y_j)$	Referenzpunkte der Units i; j	
	$(\Delta x_i, \Delta y_i); (\Delta x_j, \Delta y_j)$	Breite, Länge der Units i; j	
Binäre Variablen	$u_{ij}, v_{ij} \in \{0,1\}$ mit		
	$(u_{ij}, v_{ij}) = \left\{ \begin{array}{l} (0,0), \text{gdw. Unit i über Unit j} \\ (0,1), \text{gdw. Unit i unter Unit j} \\ (1,0), \text{gdw. Unit i rechts von Unit j} \\ (1,1), \text{gdw. Unit i links von Unit j} \end{array} \right\};$		
	im Fall F2Border		
	$(u_{ij}, v_{ij}) = \left\{ \begin{array}{l} (0,0), \text{gdw. Unit i liegt "Südseite"} \\ (0,1), \text{gdw. Unit i liegt "Nordseite"} \\ (1,0), \text{gdw. Unit i liegt "Westseite"} \\ (1,1), \text{gdw. Unit i liegt "Ostseite"} \end{array} \right\}$		

Nebenbedingungen		
Non-Negative	$x_i, y_i \in \{\mathbb{R}_+\}$ $\Delta x_i, \Delta y_i \in \{\mathbb{R}_+\}$ $x_R, y_R \in \{\mathbb{R}_+\}$ $X; Y \in \{\mathbb{R}_+\}$ $X'; Y' \in \{\mathbb{R}_+\}$	<p>Nichtnegativitätsbedingung (76)</p> <p>Nichtnegativitätsbedingung</p> <p>Nichtnegativitätsbedingung</p> <p>Nichtnegativitätsbedingung</p> <p>Nichtnegativitätsbedingung</p>
Force Inside	Die nachfolgenden Funktionen bewirken, dass Unit i vollständig innerhalb Unit j liegt:	
	$x_i \geq x_j$ $y_i \geq y_j$ $x_i + \Delta x_i - (x_j + \Delta x_j) \leq 0$ $y_i + \Delta y_i - (y_j + \Delta y_j) \leq 0$	(77)
Prohibit Intersection	Die nachfolgenden Funktionen bewirken, dass es in keiner geographischen Himmelsrichtung eine Überschneidung zwischen Unit i und Unit j gibt:	
	$x_i + \Delta x_i - x_j + u_{ij} X' + v_{ij} X' \leq 2X'$ $x_j + \Delta x_j - x_i + u_{ij} X' - v_{ij} X' \leq X'$ $y_i + \Delta y_i - y_j - u_{ij} Y' + v_{ij} Y' \leq Y'$ $y_j + \Delta y_j - y_i - u_{ij} Y' - v_{ij} Y' \leq 0$	(78)
Force Connection	Die nachfolgenden Funktionen bewirken, dass Unit i in einer beliebigen geographischen Himmelsrichtung eine Verbindung $\delta$ zu Unit j hat:	

	$y_i + \Delta y_i - y_j - u_{ij}Y' + v_{ij}Y' \leq Y'$ $y_i + \Delta y_i - y_j + u_{ij}Y' - v_{ij}Y' \geq -Y'$ $x_i + \Delta x_i - x_j + u_{ij}X' - v_{ij}X' \geq -X' + \delta$ $x_j + \Delta x_j - x_i + u_{ij}X' - v_{ij}X' \geq -X' + \delta$ $y_j + \Delta y_j - y_i - u_{ij}Y' - v_{ij}Y' \leq 0$ $y_j + \Delta y_j - y_i + u_{ij}Y' + v_{ij}Y' \geq 0$ $x_i + \Delta x_i - x_j + u_{ij}X' + v_{ij}X' \geq 0 + \delta$ $x_j + \Delta x_j - x_i + u_{ij}X' + v_{ij}X' \geq 0 + \delta$ $x_i + \Delta x_i - x_j + u_{ij}X' + v_{ij}X' \leq 2X'$ $x_i + \Delta x_i - x_j - u_{ij}X' - v_{ij}X' \geq -2X'$ $y_i + \Delta y_i - y_j - u_{ij}Y' - v_{ij}Y' \geq -2Y' + \delta$ $y_j + \Delta y_j - y_i - u_{ij}Y' - v_{ij}Y' \geq -2Y' + \delta$ $x_j + \Delta x_j - x_i + u_{ij}X' - v_{ij}X' \leq X'$ $x_j + \Delta x_j - x_i - u_{ij}X' + v_{ij}X' \geq -X'$ $y_i + \Delta y_i - y_j - u_{ij}Y' + v_{ij}Y' \geq -Y' + \delta$ $y_j + \Delta y_j - y_i - u_{ij}Y' + v_{ij}Y' \geq -Y' + \delta$ $\delta \geq 0$	(79)
Force Outside	Die nachfolgenden Funktionen bewirken, dass Unit i in einer beliebigen geographischen Himmelsrichtung außerhalb von Unit j liegt:	
	$x_i + \Delta x_i - x_j + u_{ij}(X' + \varepsilon) + v_{ij}(X' + \varepsilon) \leq 2X' + \varepsilon$ $x_j + \Delta x_j - x_i + u_{ij}(X' + \varepsilon) - v_{ij}(X' + \varepsilon) \leq X'$ $y_i + \Delta y_i - y_j - u_{ij}(Y' + \varepsilon) + v_{ij}(Y' + \varepsilon) \leq Y'$ $y_j + \Delta y_j - y_i - u_{ij}(Y' + \varepsilon) - v_{ij}(Y' + \varepsilon) \leq -\varepsilon$ $\varepsilon \geq 0$	

Force To Border	Die nachfolgenden Funktionen bewirken, dass Unit i in einer beliebigen geographischen Himmelsrichtung auf einer Innenkante von Unit j liegt:	
	$x_i + \Delta x_i - (x_j + \Delta x_j) - u_{ij} X' - v_{ij} X' \geq -2X'$ $y_i + \Delta y_i - (y_j + \Delta y_j) + u_{ij} Y' - v_{ij} Y' \geq -Y'$ $x_i - x_j + u_{ij} X' - v_{ij} X' \leq X'$ $y_i - y_j - u_{ij} Y' - v_{ij} Y' \leq 0$	(80)
Prohibit Connection	Die nachfolgenden Funktionen bewirken, dass Unit i in keiner geographischen Himmelsrichtung eine Verbindung zu Unit j hat:	
	$x_i + \Delta x_i - x_j + u_{ij} (X' + \varepsilon) + v_{ij} (X' + \varepsilon) \leq 2X' + \varepsilon$ $x_j + \Delta x_j - x_i + u_{ij} (X' + \varepsilon) - v_{ij} (X' + \varepsilon) \leq X'$ $y_i + \Delta y_i - y_j - u_{ij} (Y' + \varepsilon) + v_{ij} (Y' + \varepsilon) \leq Y'$ $y_j + \Delta y_j - y_i - u_{ij} (Y' + \varepsilon) - v_{ij} (Y' + \varepsilon) \leq -\varepsilon$ $\varepsilon > 0$	(81)
Design Constraints	<p>Length Constraint:</p> $\Delta y_i \left\{ \begin{array}{l} \leq z_i, \text{ falls Länge von Unit i nach oben begrenzt sein soll} \\ = z_i, \text{ falls Länge von Unit i genau bestimmt sein soll} \\ \geq z_i, \text{ falls Länge von Unit i nach unten begrenzt sein soll} \end{array} \right\}$ <p>Area Constraint:</p> $F_i \left\{ \begin{array}{l} \leq z_i, \text{ falls Flächeninhalt von Unit i nach oben begrenzt sein soll} \\ = z_i, \text{ falls Flächeninhalt von Unit i genau bestimmt sein soll} \\ \geq z_i, \text{ falls Flächeninhalt von Unit i nach unten begrenzt sein soll} \end{array} \right\}$	

	<p>Perimeter Constraint:</p> $U_i \left\{ \begin{array}{l} \leq z_i, \text{ falls Umfang von Unit } i \text{ nach oben begrenzt sein soll} \\ = z_i, \text{ falls Umfang von Unit } i \text{ genau bestimmt sein soll} \\ \geq z_i, \text{ falls Umfang von Unit } i \text{ nach unten begrenzt sein soll} \end{array} \right\}$ <p><math>(z_i \in \mathbb{R}_+ \text{ fest, } i \in \{1, \dots, n\})</math></p>	
Zielfunktionen		
Flächen- Maximierung	$\sum_{k \in I_1} \Delta x_k \Delta y_k \rightarrow \max$	
Umfang- Maximierung	$\sum_{k \in I_2} \Delta x_k + \Delta y_k \rightarrow \max$	
Annäherung Raum- fläche an Zielgröße	$F_0 > 0 \text{ geg. ; } (\Delta x_k \Delta y_k - F_0)^2 \rightarrow \min$	
Erzeugung quadratischer Units	$\sum_{k \in I_3} (\Delta x_k - \Delta y_k)^2 \rightarrow \min$	



## 6.2 Mathematisches Modell 2 (Zerstörungsarmes Modell)

Dieses Modell basiert auf den in Kapitel 5.3.3 beschriebenen Grundlagen. Es kann unabhängig von spezifischen Optimierungsverfahren oder Programmiersprachen verwendet werden.

### 6.2.1 Ausgangsmodell

<p>Geometriemodell</p>		<p>Nr.</p>
<p>Konventionen</p>	<p><math>G = (V, E)</math> Graph  mit  <math>V</math> Menge der Knoten</p>	

	$E \subseteq V \times V$ <i>nbreOfNodes</i> <i>Node<sub>i</sub></i> <i>nbreOfGroups</i> <i>Group<sub>j</sub></i> <i>label<sub>ij</sub></i>	Menge der Kanten Anzahl der Knoten ( $=  V $ ) Knoten $i$ des Graphen ( $i \in \{1, \dots,  V \}$ ) Anzahl der gesuchten zusammenhängenden Gruppen (Teilgraphen) Gruppe $j$ bedeutet, dass $label_i = j$	
Konstanten		keine	
Variablen			
	$A = a_{ij}$ <i>sizeNodes<sub>i</sub></i> <i>groups<sub>j</sub></i> <i>sizeGroups<sub>j</sub></i> <i>sizeRooms<sub>jk</sub></i>  <i>psu</i>  <i>pso</i>	Adjazenzmatrix von G mit $a_{ij} = \begin{cases} 1, & \text{falls } (i, j) \in E \\ 0, & \text{sonst} \end{cases}$ ; Größe von Knoten $i$ in $m^2$ Anzahl der benötigten Units in Gruppe $j$ gef. Gesamtgröße der Gruppe $j$ in $m^2$ Größe des Raumes $k$ in Gruppe $j$  ( $k \in \{1, \dots, groups_j\}$ ) Untere Schranke: Prozentsatz, welcher von der geforderten Gruppengröße mind. erfüllt sein muss (i. A. $\leq 100\%$ ) Obere Schranke: Prozentsatz, welcher von der geforderten Gruppengröße max.	

	<p>erfüllt sein darf (i. A. <math>\geq 100\%</math>)</p> <p><math>label_i</math> Gruppenzugehörigkeit des Knotens <math>i</math></p>	
Nebenbedingungen		
Anzahl	$\sum_{i=i}^{ V } label_{ij} = groups_j \quad \forall j \in \{1, \dots, nbreOfGroups\}$	(81)
Größe	$\frac{ps_o}{100} * sizeGroups_j \geq \sum_{i=i}^{ V } label_{ij} * sizeNodes_i \geq \frac{ps_u}{100} * sizeGroups_j$ $\forall j \in \{1, \dots, nbreOfGroups\}$	(82)
Zusammenhang	<p><math>\forall j \in \{1, \dots, nbreOfGroups\}</math> :</p> <p>Für alle paarweise verschiedenen Knoten <math>i, k</math></p> <p>mit <math>label_i = j, label_k = j</math></p> <p>gilt:</p> <p>entweder <math>(i, k) \in E</math></p> <p>oder es existieren <math>v_1, v_2, \dots, v_n \in V</math></p> <p>mit <math>label_{v_l} = j, l = 1..n</math>,</p> <p>so dass <math>(i, v_1), (v_1, v_2), \dots, (v_{n-1}, v_n)</math> und <math>(v_n, k) \in E</math></p>	
	<p><math>nodes</math> Bei der Breitensuche werden die gefundenen Knoten in die Variable <math>nodes</math> eingefügt, um zu prüfen, welche Knoten bereits gefunden wurden und welche neu sind.</p>	
	<p><math>anz</math> Speichert die Anzahl der gefundenen Knoten. Von Bedeutung, um neue Knoten in die Variable <math>nodes</math> einzufügen.</p>	

	<i>drin</i>	Temporäre logische Variable, die speichert, ob ein gefundener Knoten schon in der Variable <i>nodes</i> enthalten ist.	
	<i>loe</i>	Logische Variable die speichert, ob eine vom Modell gefundene Lösung tatsächlich eine Lösung des Problems darstellt.	
	<i>nbreOfSolution</i>	Speichert die Anzahl der Lösungen	
	<i>anzChecked</i>	Prüft, ob bei der Breitensuche neue, d.h. noch nicht enthaltene Knoten gefunden wurden.	
	<i>s</i>	Summe der Abweichungen für die Bestimmung der Güte	
	<i>akt</i>	Bezeichnet den Wert des aktuell größten Raumes der Gruppe. Von Bedeutung für die Bestimmung der Güte.	

## **7 Prototypen**

Aufbauend auf der mathematischen Modellbildung werden nachfolgend die drei Prototypen CATIA, VisualDOC und OPL vorgestellt. Die abschließenden Bewertungen der prototypischen Umsetzung zeigen, welches Potential und welche Probleme mit dem Einsatz der entwickelten Prototypen verbunden sind.

## 7.1 CATIA

### 7.1.1 Konzeptioneller Aufbau

Evaluiertes Einsatzbereich ZR

CATIA ist ein Softwarepaket, welches ursprünglich von der Firma Dassault entwickelt wurde und heute von IBM vertrieben wird. Es ist eine Lösung für Fertigungsunternehmen, „... die den gesamten Produktentwicklungsprozess integriert und alle Phasen des Produktlebenszyklus von der Produktkonzeption bis zum Einsatz des Produkts abdeckt...“ [IBM, 2006]. Zur Entwicklung des Prototyps wurde CATIA V5 R10 verwendet, da dieses Produkt Komponenten enthält (Knowledge Advisor, Product Engineering Optimizer), welche die regelbasierte parametrisch-assoziative Modellierung in geeigneter Weise unterstützen (vgl. Kap. 4.4).

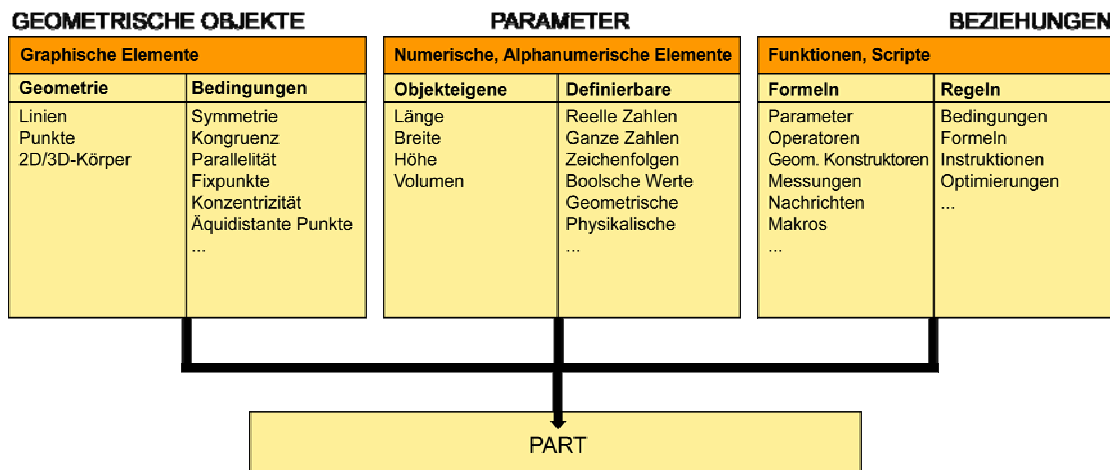


Abb. 7.1 Konzeptioneller Aufbau CATIA-Prototyp

Mit Hilfe dieser Komponenten kann die räumliche Ausprägung der Units über Parameter und Beziehungen definiert werden. Ferner beinhalten die Komponenten Optimierungsalgorithmen, welche mit Hilfe von Formeln (Zielfunktionen und Nebenbedingungen) die Parameter der Geometrie steuern können. Der Vorteil dieser Lösung ist der Zugriff auf existente Produktkomponenten, die sich im Einsatz von Fertigungsunternehmen etabliert haben.

### 7.1.2 Besonderheiten

CATIA ist von besonderem Interesse für die Umsetzung dieses Prototyps, da das Geometriemodell die Einbindung unterschiedlicher Parts (Units) erlaubt, deren Summe wiederum das Produkt (Gebäudegrundriss) repräsentiert. Auf diese Weise können die Parameter jeder einzelnen Unit gezielt miteinander verknüpft und Abhängigkeiten definiert werden.

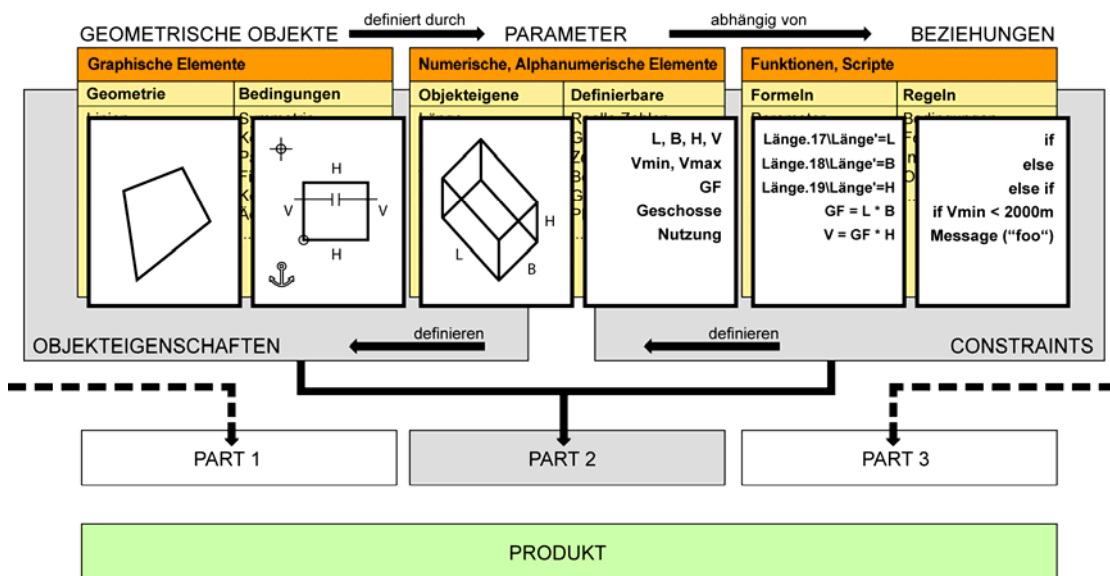


Abb. 7.2 CATIA Produktmodell

#### Aufbau des Eingabemodells

Die Steuerung der graphischen Elemente, i.e. die graphische Repräsentation einer Unit, erfolgt durch numerische und alphanumerische Elemente, welche über CATIA-Beziehungen miteinander verknüpft sind. Änderungen dieser Beziehungen wirken sich unmittelbar auf die graphische Repräsentation aus. Da die in der Product Engineering Optimizer-Komponente definierten Optimierungsbedingungen auf den CATIA-Beziehungen aufbauen, verändert eine Optimierungsrechnung synchron die Parameter der Units und somit auch die graphische Anzeige des zu optimierenden Grundrisses. Auf diese Weise lässt sich die Vorgehensweise des Algorithmus während einer Optimierungsrechnung gut nachvollziehen. Der wesentliche Vorteil dieser All-in-One Programmlösung ist der Verzicht auf externe Applikationen, welche eine potentielle Fehlersuche aufwändig gestalten. Anders als bei gängigen CAAD-Systemen ist die CATIA-Geometrie durch Constraints definiert. Eine Maßlinie ist demzufolge keine inaktive Vermaßung, sondern eine Verknüpfung eines graphischen Elements (z. B. einer Linie) mit einer Funktion (z. B. eine mathematische Formel). Diese Eigenschaft führte dazu, dass die in Kap. 6.1.1 definierten Nebenbedingungen an die Struktur der CATIA Maß-Constraints angepasst werden mussten. So benötigt bspw. das FConn Constraint bei diesem Prototyp nur eine einzige Bedingung, um in

Kombination mit den inhärenten Eigenschaften der CATIA Maß-Constraints eine Berührung zweier Units zu ermöglichen. Andererseits führten diese Eigenschaften der Maß-Constraints auch zu ungewolltem Verhalten der Units. So versuchte der Optimizer verschiedene Nebenbedingungen durch das Umstülpen der geometrischen Repräsentation der Units zu erfüllen. Um dieses Verhalten zu unterbinden, wurde ein zusätzliches Constraint definiert (Prohibit Flip-Flop). Da dieses Constraint für den allgemeinen Optimierungsprozess unbedeutend ist und lediglich eine Eigenart der CATIA-Constraints berührt, wurde auf die Darstellung dieses Constraints in den Kapiteln 5 und 6 verzichtet.

### **Aufbau des geometrischen Modells**

Das geometrische Modell des CATIA-Prototyps besteht aus 2D-Objekten in Form von Flächen. Diese Flächen entsprechen den Units und werden durch CATIA-Parameter, d.h. den Referenzpunkten und den Abständen der Referenzpunkte von den Kanten der Flächen, gebildet. Die Lage der Referenzpunkte und Kanten wird durch Maß-Constraints definiert. Die CATIA-Parameter entsprechen den Design Variablen des mathematischen Modells (vgl. Kap. 6.1.1). Numerische Bedingungen werden durch CATIA-Beziehungen definiert, welche den Response Variablen des mathematischen Modells entsprechen (vgl. Kap. 6.1.1). Diese verwenden CATIA-Parameter des geometrischen Modells (Design Variablen) und setzen sie durch die Verwendung von Formeln zueinander in Beziehung.

### **Aufbau des Optimierungsmodells**

Aus den CATIA-Parametern und -Beziehungen werden die Zielfunktionen und Nebenbedingungen zur Lösung des Optimierungsproblems definiert. Die Optimizer-Komponente von CATIA unterstützt ein Gradienten-Verfahren sowie Simulated Annealing (Simuliertes Ausglühen). Die Festlegung von Oberbereich und Unterbereich (Obere und Untere Schranken) sowie Schrittweiten ist ebenfalls möglich.

## **7.1.3 Exemplarische Ergebnisse**

Die nachfolgende Abbildung zeigt das Ergebnis einer Optimierungsrechnung in CATIA (Abb. 7.3). Evaluiert wurden hier FInside, PInter und FConn Constraints sowie eine Zielfunktion zur Erreichung bestimmter Flächengrößen der Units. Die Abbildung verdeutlicht exemplarisch die Probleme des Prototyps unter Verwendung des Simulierten Ausglühens. Der Optimizer erzielt ein lokales Optimum, welches einen großen Teil der aufgestellten Nebenbedingungen sowie die Zielfunktion erfüllt. Die Abbildung zeigt jedoch eine Verletzung des PInter Constraints der Units Rot und Orange. Das Verwerfen der erzielten Lösung, i.e. das Verlassen eines lokalen zu Gunsten eines globalen Optimums, kann unter Umständen dazu führen, dass vor Erreichen des globalen Optimums eine Lösung erzielt wird, die zunächst schlechter ist als das schon erzielte lokale Optimum. Diese Vorgehensweise wird beim Simulated Annealing Algorithmus jedoch nicht angewandt. Im Ergebnis führt dies dazu, dass bei dem Prototyp weitere Veränderungen des lokalen Optimums nur unbedeutende Veränderungen mit sich bringen. Ein globales Optimum wird nie erreicht.



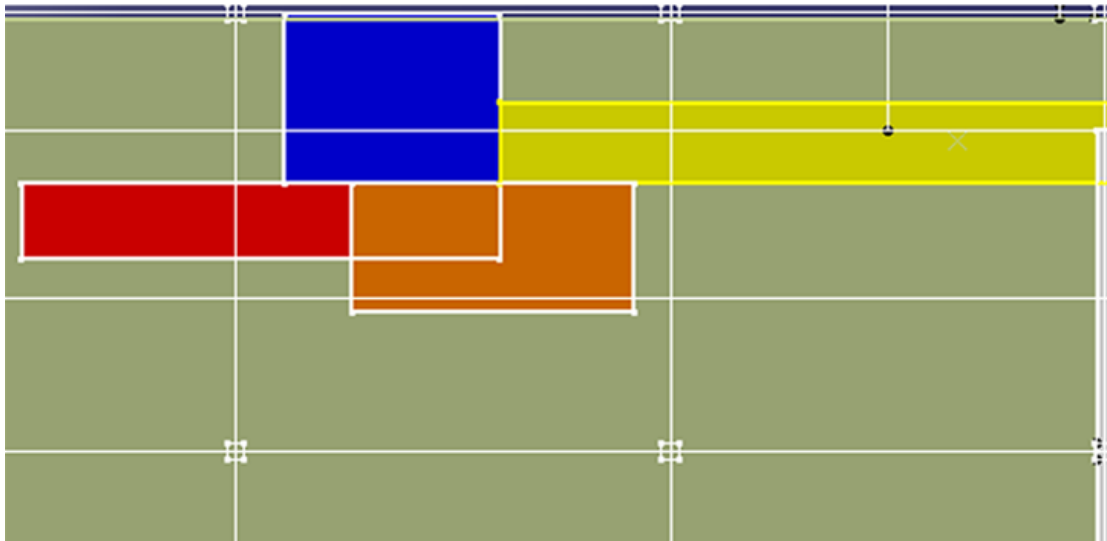


Abb. 7.3 Lokales Optimum als Ergebnis des CATIA-Prototyps

#### 7.1.4 Bewertung

Die wesentliche Schwachstelle dieses Prototyps sind die zwei in CATIA verfügbaren Optimierungsverfahren, welche für die Lösung von Problemen des Zerstörungsreichen Modells in der vorliegenden Form unzureichend sind. Das Gradienten-Verfahren konnte auf Grund des verwendeten mathematischen Modells nicht angewendet werden. Das Simulierte Ausglühen liefert im Ansatz akzeptable Ergebnisse, scheitert jedoch an den Problemen der lokalen Optima. Eine Lösung kombinatorisch aufwändiger Optimierungsprobleme kann mit diesem Prototyp nicht erreicht werden. Ein weiteres Defizit dieses Prototyps ist die fehlende Möglichkeit des benutzerdefinierten Eingriffes in die Vorgehensweise der Algorithmen. So gibt es keine Möglichkeit, die Parameter der Algorithmen zu verändern oder eigene Suchverfahren in den Optimierungsprozess zu integrieren.

## 7.2 VisualDOC

### 7.2.1 Konzeptioneller Aufbau

Evaluierter Einsatzbereich ZR

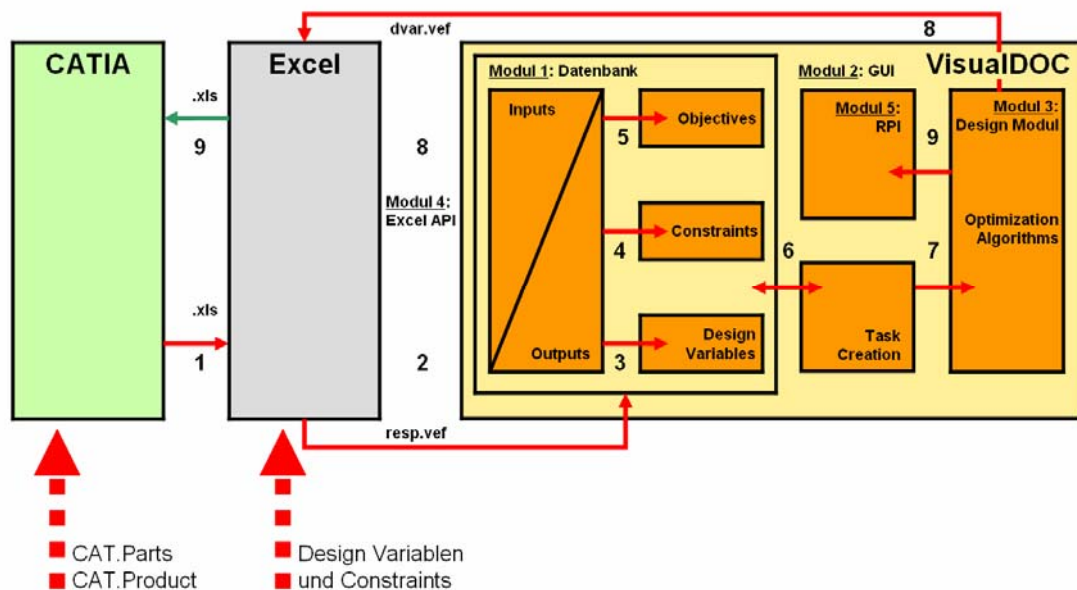


Abb. 7.4 Konzeptioneller Aufbau VisualDOC-Prototyp

VisualDOC ist ein Produkt, welches von der Firma Vanderplaats Research & Development, Colorado Springs, CO, USA vertrieben wird. Die Firma Vanderplaats bezeichnet das Programm als „Design Optimization System“ [Vanderplaats, 2006]. Auf Grund der sehr komplexen Bedeutung des englischen Wortes „Design“ ist es schwierig, eine vergleichbare deutsche Übersetzung für den Einsatzbereich des Produktes zu definieren. Die von Vanderplaats vorgestellten Anwendungsbereiche sind im Wesentlichen Problemklassen der linearen und nichtlinearen Optimierung. Es werden sowohl Optimierungsfälle für chemische Prozesse als auch für mechanische, bauklimatische und statische Berechnungen demonstriert. Die zur Verfügung stehenden Verfahren bedienen sich gradienten-basierter, nichtgradienten-basierter und probabilistischer Algorithmen sowie der Verfahren „Response Surface Approximation“ und „Design of Experiments“. Das Produkt besteht aus 5 Modulen zur Definition und Berechnung von Optimierungsproblemen. Alle Eingabedaten des Benutzers (z. B. Variablen) werden in einer VisualDOC-Datenbank gespeichert (Modul 1), auf welche mit einer graphischen Benutzeroberfläche zugegriffen wer-

den kann (Modul 2). Das Design Modul beinhaltet die zur Berechnung des Optimierungsproblems notwendigen Algorithmen (Modul 3). Zur Interaktion der Datenbank mit den vom Benutzer in einem externen Analyseprogramm zu spezifizierenden mathematischen Formeln der Design Variablen, Nebenbedingungen und Zielfunktionen, werden verschiedene API's zur Verfügung gestellt (Modul 4). Es existieren ferner Schnittstellen zu Excel und Matlab, so dass Design Variablen mit Hilfe dieser Applikationen repräsentiert werden können. Zur Auswertung der Berechnungsergebnisse steht ein so genanntes „Response Program Interface (RPI)“ zur Verfügung, das Berechnungsergebnisse in Form von Diagrammen graphisch auswerten kann (Modul 5). Der konzeptionelle Aufbau des entwickelten Prototyps besteht aus einer CATIA-Konstruktionstabelle, welche die graphische Aufbereitung der numerischen Berechnungsergebnisse ermöglicht. Die Originaldaten der Konstruktionstabelle, d.h. die Ausgangspositionen der Units werden in Excel eingelesen und im Fall von Design Variablen, Nebenbedingungen und Zielfunktionen mathematisch definiert (Abb. 7.4, Nr.1). Über die Excel API werden diese Daten automatisch in VisualDOC importiert und dort zur Definition der Optimierungsaufgabe (Abb. 7.4, Nr. 2-6) verwendet. Diese Optimierungsaufgabe wird in der VisualDOC-Datenbank abgelegt und zur Berechnung an das Design Modul weitergeleitet (Abb. 7.4, Nr. 7). Die Berechnung der Ergebnisse erfolgt bidirektional unter ständiger Rückkopplung mit Excel (Abb. 7.4, Nr. 8 und 2). Das Endergebnis wird in der Excel-Tabelle gespeichert und entweder über das RPI diagrammatisch ausgewertet oder an die Konstruktionstabelle von CATIA zur graphischen Darstellung des Gebäudegrundrisses weitergeleitet (Abb. 7.4, Nr. 9).

## 7.2.2 Besonderheiten

Die Besonderheiten dieses Prototyps liegen in der Verwendung von drei unterschiedlichen Programmen zur Berechnung und Anzeige des Optimierungsproblems. VisualDOC verlangt ein externes Analyse-Programm, welches zur Definition der Design Variablen, Nebenbedingungen und Zielfunktionen verwendet werden muss. Diese Konstellation stellte sich im Verlauf der Evaluation als Schwachstelle heraus, da insbesondere die Schnittstelle zu Excel nicht fehlerfrei ist. So zeigt bspw. die Evaluation des FConn-Constraints (vgl. Tab. 7.10 - Tab. 7.13) divergierende Werte der Excel-Tabelle und der Report-Datei von VisualDOC. Aus diesem Grund wurde zusätzlich ein selbst geschriebenes Analyse-Programm unter Verwendung der C++ API eingesetzt (vgl. Kap. 7.2.4). Die Verwendung von CATIA zur graphischen Anzeige war notwendig, da eine Anzeige der Optimierungsergebnisse in VisualDOC nur in diagrammatischer oder numerischer Form möglich ist. Komplexe Berechnungen mit einer großen Anzahl von Variablen können auf diese Weise jedoch nicht überschaubar dargestellt werden.

Abweichend von der Standardform eines Optimierungsmodells (vgl. Kap. 5.3.1), erlaubt es VisualDOC einen einzigen Wert (Target) als Zielfunktion zu definieren. Ferner ist es möglich, Optimierungsprobleme mit mehrfacher - auch divergierender - Zielsetzung (z. B. minimiere  $F_i$ , maximiere  $F_j$ ) zu definieren (Multi-Objective Optimization).

### 7.2.3 Exemplarische Ergebnisse

Die nachfolgenden Beispiele zeigen die Anwendung des Prototyps auf rechteckige Gebäudegrundrisse und Units des Zerstörungsreichen Modells.

Anhand der Beispiele wurde die grundsätzliche Funktionsfähigkeit des mathematischen Modells untersucht (vgl. Kap. 6.1.1). Als Zielfunktion wurden dabei zunächst lediglich Targets definiert. Die Nebenbedingungen wurden zur detaillierten Evaluierung des mathematischen Modells jeweils separat betrachtet. Es wurden ferner unterschiedliche Algorithmen (SQP – Sequential Quadratic Programming, SLP – Sequential Linear Programming, MMFD – Modified Method of Feasible Directions, SUO – Sequential Unconstrained Optimization) und deren Anwendung zur Lösung von Optimierungsproblemen des Zerstörungsreichen Modells evaluiert. Die Beispiele 1-6 belegen die Funktionsfähigkeit der Constraints Force Inside, Prohibit Intersection, Force Connection, Force Outside, Force To Border und Prohibit Connection. In den Beispielen wird die Grundunit R mit j bezeichnet.

## Beispiel 1

### Datei

siehe Tabelle (Tab. 7.2 - Tab. 7.4)

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.1)

### Zielfunktion

Target (Größe der Flächen  $F_f = 20\text{m}^2$ ,  $F_g = 20\text{m}^2$ ,  $F_h = 20\text{m}^2$ ,  $F_i = 20\text{m}^2$ )

### Nebenbedingungen

Als Nebenbedingung wird ausschließlich das Force Inside Constraint evaluiert.

### Berechnungszeit

siehe Tabelle (Tab. 7.2 - Tab. 7.4)

### Lösung

Bei der Evaluation des Force Inside Constraints wurde häufig die unvollständige Erfüllung zu maximierender Zielfunktionen beobachtet. Der Grund dafür lag in der Verwendung Unterer Schranken (Worst Value), welche innerhalb zu maximierender Zielfunktionen die Minimalwerte der Zielfunktionserfüllung vorgeben. Obwohl diese Berechnungen mit Erfüllung der Kuhn-Tucker-Konditionen beendet wurden, d.h. ein globales Optimum hätte erreicht sein müssen, zeigen Beispielrechnungen ohne Verwendung von Worst Values bessere Werte der Zielfunktionserfüllung. Die Verwendung von Worst Values innerhalb der Zielfunktionen ist daher nicht empfehlenswert.

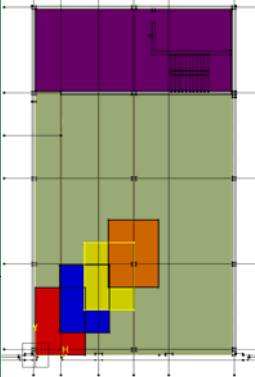
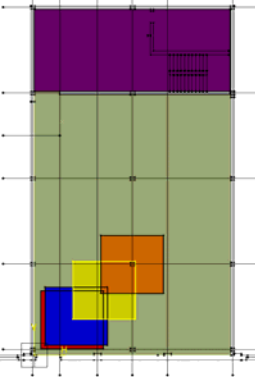
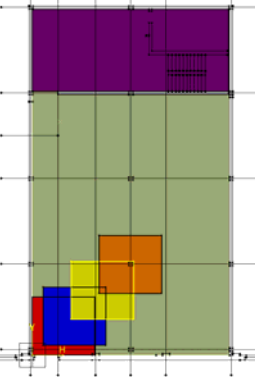
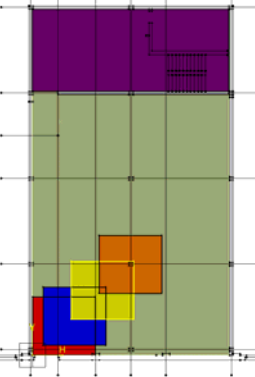
Die nachfolgenden Tabellen verdeutlichen, dass mit Ausnahme des SQP-Algorithmus alle Algorithmen die vollständige Erfüllung der Zielfunktionen erreichen. Die Nebenbedingungen des Optimierungsproblems werden von allen Algorithmen vollständig erfüllt.

Force Inside	Erläuterung	Globaler Index (Constraints)
Task-Präfix:	FInside_	<b>Farbschemata:</b>
Task-Nummer:	exp_1	sehr gut
VisualDoc Task-Bezeichnung:	FInside_exp_1	befriedigend
Task Type / Constrained Method / Unconstrained Method / Gradient		schlecht
Task:	<p>Flächen orange, gelb, blau, rot (f, g, h, i) innerhalb der Grundunit beige (j) anordnen</p> <p>Overall Multiobjective Optimization Goal: Minimize</p> <p>Stopping Criteria:                      Relative Objective Convergence: 1.00E-3                      Absolute Objective Convergence: 1.00E-6</p> <p>Max. Iterations: 100, Iterations for Convergence: 10</p>	<p><b>Akronyme: Constrained Method</b>                      SQP - Sequential Quadratic Programming                      SLP - Sequential Linear Programming                      MMFD - Modified Method of Feasible Directions                      SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Unconstrained Method</b>                      BFGS - Broydon-Fletcher-Goldfarb-Shanno                      FR - Fletcher-Reeves                      SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Gradients</b>                      CD - Central Difference                      FD - Forward Difference</p>
Constraints:	<p><b>FInside</b>                      Umsetzung durch Lower und Upper Bounds der Flächen (f, g, h, i). Lower und Upper Bounds dieser Units entsprechen den konstanten x- und y-Werten der Grundunit beige (j).</p> <p>Finside sollte jedoch auch mit folgendem Constraint umgesetzt werden (siehe FInside_exp_2):</p> <p><b>Force Inside (i, j) == (i inside j)</b>  <math>MAX(Y_{Ni} - Y_{Nj}, Y_{Sj} - Y_{Si}, X_{Ei} - X_{Ej}, X_{Wj} - X_{Wi}) \leq 0</math></p>	
Constraint-Erfüllung:		
Objectives: (VisualDOC Task Report)	Ff = 20qm, Fg = 20qm, Fh = 20qm, Fi = 20qm	
Runtime:		
VisualDoc-Excel-Output: File-Name		
Visualization:		
Summary-File:		
Bemerkungen:		

Tab. 7.1 Force Inside

	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	Finside_exp1_gradient_sqp_bfqs_cd gradient / sqp / bfqs / central difference	Finside_exp1_gradient_slp_bfqs_cd gradient / slp / bfqs / central difference	Finside_exp1_gradient_mmfd_bfqs_cd gradient / mmfd / bfqs / central difference	Finside_exp1_gradient_suo_bfqs_cd gradient / suo / bfqs / central difference
	ja	ja	ja	ja
	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !
	Alle Flächen ~ 18,54 qm Objectives nur annähernd erfüllt !	2 Flächen 19,99 qm; 2 Flächen 20,00 qm Objectives vollständig erfüllt !	Alle Flächen 19,99 qm Objectives vollständig erfüllt !	3 Flächen 19,99 qm; eine Fläche 20,00 qm Objectives vollständig erfüllt !
	00:00:19	00:12:53	00:01:02	00:15:14
	Finside_exp_1_gradient_sqp_bfqs_cd.xls	Finside_exp_1_gradient_slp_bfqs_cd.xls	Finside_exp_1_gradient_mmfd_bfqs_cd.xls	Finside_exp_1_gradient_suo_bfqs_cd.xls
	Finside_exp_1_gradient_sqp_bfqs_cd.tif	Finside_exp_1_gradient_slp_bfqs_cd.tif	Finside_exp_1_gradient_mmfd_bfqs_cd.tif	Finside_exp_1_gradient_suo_bfqs_cd.tif
	Finside_exp_1_gradient_sqp_bfqs_cd.out	Finside_exp_1_gradient_slp_bfqs_cd.out	Finside_exp_1_gradient_mmfd_bfqs_cd.out	Finside_exp_1_gradient_suo_bfqs_cd.out
	Zero search direction in sqp subproblem	Maximum number of iterations reached	Relative objective convergence criterion met	Kuhn-Tucker conditions met

Tab. 7.2 Force Inside BFGS

	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	Finside_exp1_gradient_sqp_fr_cd	Finside_exp1_gradient_slp_fr_cd	Finside_exp1_gradient_mmfd_fr_cd	Finside_exp1_gradient_suo_fr_cd
	gradient / sqp / fr / central difference	gradient / slp / fr / central difference	gradient / mmfd / fr / central difference	gradient / suo / fr / central difference
	ja	ja	ja	ja
				
	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !
	Alle Flächen = 18,54 qm Objectives nur annähernd erfüllt !	2 Flächen 18,89 qm; 2 Flächen 20,00 qm Objectives vollständig erfüllt !	Alle Flächen 19,99 qm Objectives vollständig erfüllt !	3 Flächen 19,99 qm; eine Fläche 20,00 qm Objectives vollständig erfüllt !
	00:00:16	00:13:29	00:01:01	00:15:28
	Finside_exp_1_gradient_sqp_fr_cd.xls	Finside_exp_1_gradient_slp_fr_cd.xls	Finside_exp_1_gradient_mmfd_fr_cd.xls	Finside_exp_1_gradient_suo_fr_cd.xls
	Finside_exp_1_gradient_sqp_fr_cd.tif	Finside_exp_1_gradient_slp_fr_cd.tif	Finside_exp_1_gradient_mmfd_fr_cd.tif	Finside_exp_1_gradient_suo_fr_cd.tif
	Finside_exp_1_gradient_sqp_fr_cd.out	Finside_exp_1_gradient_slp_fr_cd.out	Finside_exp_1_gradient_mmfd_fr_cd.out	Finside_exp_1_gradient_suo_fr_cd.out
	Zero search direction in SQP sub-problem	Maximum number of iterations reached	Relative objective convergence criterion met	Kuhn-Tucker conditions met

Tab. 7.3 Force Inside FR



	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	Finside_exp1_gradient_sqp_suo_cd	Finside_exp1_gradient_slp_suo_cd	Finside_exp1_gradient_mmfd_suo_cd	Finside_exp1_gradient_suo_suo_cd
	gradient / sqp / suo / central difference	gradient / slp / suo / central difference	gradient / mmfd / suo / central difference	gradient / suo / suo / central difference
	ja	ja	ja	ja
	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !
	Alle Flächen = 18,54 qm Objectives nur annähernd erfüllt !	2 Flächen 18,89 qm; 2 Flächen 20,00 qm Objectives vollständig erfüllt !	Alle Flächen 19,99 qm Objectives vollständig erfüllt !	3 Flächen 19,99 qm; eine Fläche 20,00 qm Objectives vollständig erfüllt !
	00:00:17	00:12:54	00:01:08	00:15:43
	Finside_exp 1 gradient_sqp_suo_cd.xls	Finside_exp 1 gradient_slp_suo_cd.xls	Finside_exp 1 gradient_mmfd_suo_cd.xls	Finside_exp 1 gradient_suo_suo_cd.xls
	Finside_exp 1 gradient_sqp_suo_cd.tif	Finside_exp 1 gradient_slp_suo_cd.tif	Finside_exp 1 gradient_mmfd_suo_cd.tif	Finside_exp 1 gradient_suo_suo_cd.tif
	Finside_exp 1 gradient_sqp_suo_cd.out	Finside_exp 1 gradient_slp_suo_cd.out	Finside_exp 1 gradient_mmfd_suo_cd.out	Finside_exp 1 gradient_suo_suo_cd.out
	Zero search direction in SQP sub-problem	Maximum number of iterations reached	Relative objective convergence criterion met	Kuhn-Tucker conditions met

Tab. 7.4 Force Inside SUO

## Beispiel 2

### Datei

siehe Tabelle (Tab. 7.6 - Tab. 7.8)

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.1)

### Zielfunktion

Target (Größe der Flächen  $F_f = 20\text{m}^2$ ,  $F_g = 20\text{m}^2$ ,  $F_h = 20\text{m}^2$ ,  $F_i = 20\text{m}^2$ )

### Nebenbedingungen

Als Nebenbedingung wird ausschließlich das Prohibit Intersection Constraint evaluiert.

### Berechnungszeit

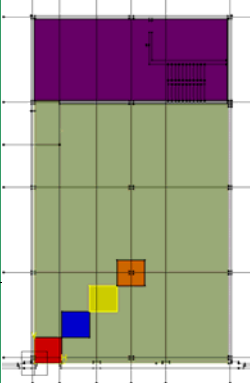
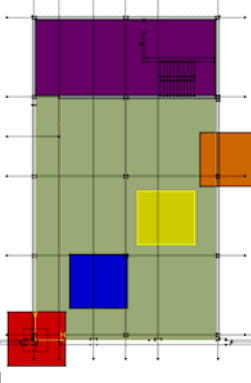
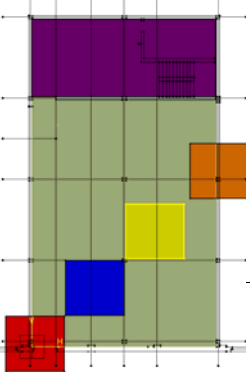
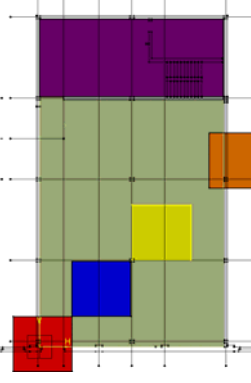
siehe Tabelle (Tab. 7.6 - Tab. 7.8)

### Lösung

Die nachfolgenden Tabellen verdeutlichen, dass mit Ausnahme des SQP-Algorithmus alle Algorithmen die vollständige Erfüllung der Zielfunktionen erreichen. Die Nebenbedingungen des Optimierungsproblems werden von allen Algorithmen vollständig erfüllt.

Prohibit Intersection		Globaler Index (Constraints)	
Task-Präfix:	PInter_	<b>Farbschemata:</b>	
Task-Nummer:	exp_1	sehr gut	
VisualDoc Task-Bezeichnung:	PInter_exp_1	befriedigend	
Task Type / Constrained Method / Unconstrained Method / Gradient		schlecht	
Task:	<p>Flächenorgane gelb, blau, rot (f, g, h, i) ohne Überschneidungen anordnen. Die Flächen müssen nicht innerhalb der Grundunit beige (j) liegen. Daher gibt keine Beschränkungen der Lower und Upper Bounds des Finside-Constraints. Somit ist Finside deaktiviert und ausschließlich PInter_exp_1 wird evaluiert.</p> <p>Da die PInter-Constraints erfüllt sind, wenn der kleinste Wert kleiner oder gleich Null ist, dürfen die Upper Bounds der PInter-Constraints nicht größer als Null sein. Da der kleinste Wert theoretisch unendlich klein sein kann (eine Unit ist unendlich weit von einer anderen entfernt), sind die Lower Bounds auf NONE gesetzt. Diese zwei Werte bedingen, dass sich die Flächen nicht überschneiden können.</p> <p>Overall Multiobjective Optimization Goal: Minimize</p> <p>Stopping Criteria: Relative Objective Convergence: 1.00E-3 Absolute Objective Convergence: 1.00E-6</p> <p>Max. Iterations: 100, Iterations for Convergence: 10</p>	<p><b>Akronyme: Constrained Method</b> SQP - Sequential Quadratic Programming SLP - Sequential Linear Programming MMFD - Modified Method of Feasible Directions SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Unconstrained Method</b> BFGS - Broydon-Fletcher-Goldfarb-Shanno FR - Fletcher-Reeves SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Gradients</b> CD - Central Difference FD - Forward Difference</p>	
Constraints:	<p><b>PInter</b> Umsetzung durch PInter_Constraint der Flächen (f, g, h, i)</p> <p><b>Prohibit Intersection (f, g) == (f north/south/east/west of g)</b>  <math>\text{MIN}(Y_{Ng} - Y_{Sf}, Y_{Nf} - Y_{Sg}, X_{Eg} - X_{Wf}, X_{Ef} - X_{Wg}) \leq 0</math></p> <p><b>Prohibit Intersection (f, h) == (f north/south/east/west of h)</b>  <math>\text{MIN}(Y_{Nh} - Y_{Sf}, Y_{Nf} - Y_{Sh}, X_{Eh} - X_{Wf}, X_{Ef} - X_{Wh}) \leq 0</math></p> <p><b>Prohibit Intersection (f, i) == (f north/south/east/west of i)</b>  <math>\text{MIN}(Y_{Ni} - Y_{Sf}, Y_{Nf} - Y_{Si}, X_{Ei} - X_{Wf}, X_{Ef} - X_{Wi}) \leq 0</math></p> <p><b>Prohibit Intersection (g, h) == (g north/south/east/west of h)</b>  <math>\text{MIN}(Y_{Nh} - Y_{Sg}, Y_{Ng} - Y_{Sh}, X_{Eh} - X_{Wg}, X_{Eg} - X_{Wh}) \leq 0</math></p> <p><b>Prohibit Intersection (g, i) == (g north/south/east/west of i)</b>  <math>\text{MIN}(Y_{Ni} - Y_{Sg}, Y_{Ng} - Y_{Si}, X_{Ei} - X_{Wg}, X_{Eg} - X_{Wi}) \leq 0</math></p> <p><b>Prohibit Intersection (h, i) == (h north/south/east/west of i)</b>  <math>\text{MIN}(Y_{Ni} - Y_{Sh}, Y_{Nh} - Y_{Si}, X_{Ei} - X_{Wh}, X_{Eh} - X_{Wi}) \leq 0</math></p>		
Constraint-Erfüllung:			
Objectives: (VisualDOC Task Report)	Fi = 20qm, Fh = 20qm, Fg = 20qm, Fi = 20qm		
Runtime:			
VisualDoc-Excel-Output: File-Name			
Visualization:			
Summary-File:			
Bemerkungen:			

Tab. 7.5 Prohibit Intersection

	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	Pinter_exp1_gradient_sqp_bfgs_cd	Pinter_exp1_gradient_slp_bfgs_cd	Pinter_exp1_gradient_mmfd_bfgs_cd	Pinter_exp1_gradient_suo_bfgs_cd
	gradient / sqp / bfgs / central difference	gradient / slp / bfgs / central difference	gradient / mmfd / bfgs / central difference	gradient / suo / bfgs / central difference
	nein	ja	ja	ja
				
	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !
	FF=20.00 qm FH=20.00 qm Fg=4.00 qm FH=4.00 qm Kuhn-Objective erfüllt !	FF=20.00 qm FH=20.00 qm Fg=20.01 qm FH=20.00 qm Objectives vollständig erfüllt !	Alle Flächen 20.00 qm Objectives vollständig erfüllt !	Alle Flächen 20.00 qm Objectives vollständig erfüllt !
	00:00:09	00:11:51	00:03:43	00:25:10
BFGS Broyden-Fletcher-Goldfarb-Shanno	Pinter_exp_1_gradient_sqp_bfgs_cd.xls	Pinter_exp_1_gradient_slp_bfgs_cd.xls	Pinter_exp_1_gradient_mmfd_bfgs_cd.xls	Pinter_exp_1_gradient_suo_bfgs_cd.xls
	Pinter_exp_1_gradient_sqp_bfgs_cd.tif	Pinter_exp_1_gradient_slp_bfgs_cd.tif	Pinter_exp_1_gradient_mmfd_bfgs_cd.tif	Pinter_exp_1_gradient_suo_bfgs_cd.tif
	Pinter_exp_1_gradient_sqp_bfgs_cd.out	Pinter_exp_1_gradient_slp_bfgs_cd.out	Pinter_exp_1_gradient_mmfd_bfgs_cd.out	Pinter_exp_1_gradient_suo_bfgs_cd.out
	Zero search direction in SQP sub-problem	Maximum number of iterations reached	Relative objective convergence criterion met	Kuhn-Tucker conditions met

Tab. 7.6 Prohibit Intersection BFGS

	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	Pinter_exp1_gradient_sqp_fr_cd	Pinter_exp1_gradient_slp_fr_cd	Pinter_exp1_gradient_mmfd_fr_cd	Pinter_exp1_gradient_suo_fr_cd
	gradient / sqp / fr / central difference	gradient / slp / fr / central difference	gradient / mmfd / fr / central difference	gradient / suo / fr / central difference
	nein	ja	ja	ja
	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !
	Alle Flächen 4,00 qm Raus! Objective erfüllt !	F=20,00 qm FH=20,00 qm Fg=20,01 qm FF=20,00 qm Objectives vollständig erfüllt !	Alle Flächen 20,00 qm Objectives vollständig erfüllt !	Alle Flächen 20,00 qm Objectives vollständig erfüllt !
	00:00:10	00:11:43	00:02:23	00:24:51
	Pinter_exp_1_gradient_sqp_fr_cd.xls	Pinter_exp_1_gradient_slp_fr_cd.xls	Pinter_exp_1_gradient_mmfd_fr_cd.xls	Pinter_exp_1_gradient_suo_fr_cd.xls
	Pinter_exp_1_gradient_sqp_fr_cd.tif	Pinter_exp_1_gradient_slp_fr_cd.tif	Pinter_exp_1_gradient_mmfd_fr_cd.tif	Pinter_exp_1_gradient_suo_fr_cd.tif
	Pinter_exp_1_gradient_sqp_fr_cd.out	Pinter_exp_1_gradient_slp_fr_cd.out	Pinter_exp_1_gradient_mmfd_fr_cd.out	Pinter_exp_1_gradient_suo_fr_cd.out
	Zero search direction in SQP sub-problem	Maximum number of iterations reached	Relative objective convergence criterion met	Kuhn-Tucker conditions met

Tab. 7.7 Prohibit Intersection FR

	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	Pinter_exp1_gradient_sqp_suo_cd	Pinter_exp1_gradient_slp_suo_cd	Pinter_exp1_gradient_mmfd_suo_cd	Pinter_exp1_gradient_suo_suo_cd
	gradient / sqp / suo / central difference	gradient / slp / suo / central difference	gradient / mmfd / suo / central difference	gradient / suo / suo / central difference
	nein	ja	ja	ja
	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !	Alle Constraints vollständig erfüllt !
	Alle Flächen 4,00 qm Rest Objective erfüllt !	F=20,00 qm FH=20,00 qm Fg=20,01 qm FF=20,00 qm Objectives vollständig erfüllt !	Alle Flächen 20,00 qm Objectives vollständig erfüllt !	Alle Flächen 20,00 qm Objectives vollständig erfüllt !
	00:00:10	00:11:39	00:02:25	00:24:30
	Pinter_exp_1_gradient_sqp_suo_cd.xls	Pinter_exp_1_gradient_slp_suo_cd.xls	Pinter_exp_1_gradient_mmfd_suo_cd.xls	Pinter_exp_1_gradient_suo_suo_cd.xls
	Pinter_exp_1_gradient_sqp_suo_cd.tif	Pinter_exp_1_gradient_slp_suo_cd.tif	Pinter_exp_1_gradient_mmfd_suo_cd.tif	Pinter_exp_1_gradient_suo_suo_cd.tif
	Pinter_exp_1_gradient_sqp_suo_cd.out	Pinter_exp_1_gradient_slp_suo_cd.out	Pinter_exp_1_gradient_mmfd_suo_cd.out	Pinter_exp_1_gradient_suo_suo_cd.out
	Zero search direction in SQP sub-problem	Maximum number of iterations reached	Relative objective convergence criterion met	Kuhn-Tucker conditions met
SUO Sequential Unconstrained Optimization (BIGDOT)				

Tab. 7.8 Prohibit Intersection SUO

### Beispiel 3

#### Datei

siehe Tabelle (Tab. 7.10 - Tab. 7.13)

#### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.1)

#### Zielfunktion

Target (Größe der Flächen  $F_f = 20\text{m}^2$ ,  $F_g = 20\text{m}^2$ ,  $F_h = 20\text{m}^2$ ,  $F_i = 20\text{m}^2$ )

#### Nebenbedingungen

Als Nebenbedingung wird ausschließlich das Force Connection Constraint evaluiert.

#### Berechnungszeit

siehe Tabelle (Tab. 7.10 - Tab. 7.13)

#### Lösung

Die nachfolgenden Tabellen verdeutlichen, dass mit Ausnahme des SQP-Algorithmus alle Algorithmen die vollständige Erfüllung der Zielfunktionen erreichen. Die Nebenbedingungen des Optimierungsproblems werden jedoch weder vom SQP-Algorithmus noch vom SUO-Algorithmus vollständig erfüllt. Auf Grund des unbefriedigenden Ergebnisses der SQP- und SUO-Algorithmen wurden zusätzliche Experimente mit genetischen Algorithmen (GA) durchgeführt. Exemplarisch verdeutlicht Tab. 7.13 die ebenfalls unbefriedigende Performance der GA's.

Force Connection		Globaler Index (Constraints)	
Task-Präfix:	FConn_	<b>Farbschemata:</b>	
Task-Nummer:	exp_1	sehr gut	
VisualDoc Task-Bezeichnung:	FConn_exp_1	befriedigend	
Task Type / Constrained Method / Unconstrained Method / Gradient		schlecht	
Task:	<p>Flächen crimson, orange, blau, rot (e, f, h, i) mit Verbindung zur Fläche gelb (g) anordnen. Alle Flächen müssen die Fläche gelb (g) berühren. Als Mindestmaß für die Berührung wird durch Upper- und Lower Bounds für das jeweils 2. FConn-Constraint (z.B. FConnN2) 1m festgelegt. Diese Mindestberührung sichert eine mögliche Verbindung der Flächen in der Breite eines Verkehrsweges. Die Flächen müssen nicht innerhalb der Grundunit beige (j) liegen. Daher gibt keine Beschränkungen der Lower und Upper Bounds des Finside-Constraints. Die Flächen dürfen sich überschneiden. Somit sind Finside und Pinter deaktiviert und ausschließlich FConn_exp_1 wird evaluiert.</p> <p>Lower und Upper Bounds des 1. Constraints sind Null. Lower Bounds des 2. Constraints sind 1m, dies entspricht der Mindestbreite eines Verkehrsweges. Die Upper Bounds des 2. Constraints können gegen unendlich streben und sind deshalb auf NONE gesetzt. Da für den Berührungsbereich der jeweils kleinste Wert (Min-Funktion) maßgebend ist, spielt der Wert des größeren Terms keine Rolle.</p> <p>Overall Multiobjective Optimization Goal: Minimize</p> <p>Stopping Criteria: Relative Objective Convergence: 1.00E-3</p>	<p><b>Akronyme: Constrained Method</b> SQP - Sequential Quadratic Programming SLP - Sequential Linear Programming MMFD - Modified Method of Feasible Directions SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Unconstrained Method</b> BFGS - Broydon-Fletcher-Goldfarb-Shanno FR - Fletcher-Reeves SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Gradients</b> CD - Central Difference FD - Forward Difference</p>	
Constraints:	<p><b>FConn</b> Umsetzung durch FConn_Constraint der Flächen (e, f, h, i). <b>Force Connection (e, g) == (e north of g)</b> FConnN1 (e, g): <math>MIN(X_{Ng} - Y_{Se}) \geq 0</math> FConnN2 (e, g): <math>MIN(X_{Eg} - X_{We}; X_{Ee} - X_{Wg}) \geq 0</math> <b>Force Connection (f, g) == (f south of g)</b> FConnS1 (f, g): <math>MIN(Y_{Nf} - Y_{Sg}) \geq 0</math> FConnS2 (f, g): <math>MIN(X_{Eg} - X_{Wf}; X_{Ef} - X_{Wg}) \geq 0</math> <b>Force Connection (h, g) == (h east of g)</b> FConnE1 (h, g): <math>X_{Eg} - X_{Wh} = 0</math> FConnE2 (h, g): <math>MIN(Y_{Ng} - Y_{Sh}; Y_{Nh} - Y_{Sg}) \geq 0</math> <b>Force Connection (i, g) == (i west of g)</b> FConnW1 (i, g): <math>X_{Ei} - X_{Wg} = 0</math> FConnW2 (i, g): <math>MIN(Y_{Ni} - Y_{Si}; Y_{Ni} - Y_{Sg}) \geq 0</math></p>		
Constraint-Erfüllung:			
Objectives: (VisualDOC Task Report)	Fe = 20qm, Ff = 20qm, Fg = 20qm, Fh = 20qm, Fi = 20qm		
Runtime:			
VisualDoc-Excel-Output: File-Name			
Visualizaton:			
Summary-File:			
Bemerkungen:			

Tab. 7.9 Force Connection



	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	FConn_exp1_gradient_sqp_bfqs_cd	FConn_exp1_gradient_slp_bfqs_cd	FConn_exp1_gradient_mmfd_bfqs_cd	FConn_exp1_gradient_suo_bfqs_cd
	gradient / sqp / bfqs / central difference	gradient / slp / bfqs / central difference	gradient / mmfd / bfqs / central difference	gradient / suo / bfqs / central difference
	nein	ja	ja	ja
	Kein Constraints erfüllt!	Alle Constraints vollständig erfüllt!	Alle Constraints vollständig erfüllt!	3 verletzte Constraints (max. 8 cm)
	Kein Flächen 4,00 qm Kein Objective erfüllt!	3 Flächen 20,00 qm; 2 Flächen 19,99 qm Objectives vollständig erfüllt!	Alle Flächen 20,00 qm Objectives vollständig erfüllt!	F=20,03 qm Fh=20,02 qm Fg u. Ff=20,00 qm Fe=19,87 qm Objectives vollständig erfüllt!
	00:00:12	00:29:35	00:12:48	01:05:07
	FConn_exp_1_gradient_sqp_bfqs_cd.xls	FConn_exp_1_gradient_slp_bfqs_cd.xls	FConn_exp_1_gradient_mmfd_bfqs_cd.xls	FConn_exp_1_gradient_suo_bfqs_cd.xls
	FConn_exp_1_gradient_sqp_bfqs_cd.tif	FConn_exp_1_gradient_slp_bfqs_cd.tif	FConn_exp_1_gradient_mmfd_bfqs_cd.tif	FConn_exp_1_gradient_suo_bfqs_cd.tif
	FConn_exp_1_gradient_sqp_bfqs_cd.out	FConn_exp_1_gradient_slp_bfqs_cd.out	FConn_exp_1_gradient_mmfd_bfqs_cd.out	FConn_exp_1_gradient_suo_bfqs_cd.out
	No feasible solution found	Maximum number of iterations reached	Relative objective convergence criterion met	Maximum number of iterations reached

**ANMERKUNG:** Zeile K1-K54 der Datei: FConn\_exp\_1\_gradient\_suo\_bfqs\_cd.xls zeigt die Ergebnisse der VisualDOC-Berechnung. Die Daten wurden aus der folgenden Report-Datei von VisualDOC entnommen:  
FConn\_exp\_1\_gradient\_suo\_bfqs\_cd.out  
Das Excel-Interface zeigte andere Werte (B1-B54), die nicht mit den von VisualDOC errechneten Werten übereinstimmen. Auf Grund der richtigen Berechnungsergebnisse des Optimizers wurden dessen Werte als Grundlage der Bewertung in die Gesamtergebnisse übernommen.

Tab. 7.10 Force Connection BFGS

	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	FConn_exp1_gradient_sqp_fr_cd	FConn_exp1_gradient_slp_fr_cd	FConn_exp1_gradient_mmfd_fr_cd	FConn_exp1_gradient_suo_fr_cd
	gradient / sqp / fr / central difference	gradient / slp / fr / central difference	gradient / mmfd / fr / central difference	gradient / suo / fr / central difference
	nein	ja	ja	ja
	Kein Constraint erfüllt!	Alle Constraints vollständig erfüllt!	Alle Constraints vollständig erfüllt!	4 verletzte Constraints (max. 8 cm)
	Kein Flächen=100 qm Kein Objective erfüllt!	3 Flächen 20,00 qm; 2 Flächen 19,99 qm Objectives vollständig erfüllt!	Alle Flächen 20,00 qm Objectives vollständig erfüllt!	FF=20,03 qm FH=20,02 qm Fg u. FH=20,00 qm Fe=19,87 qm Objectives vollständig erfüllt!
	00:00:17	00:34:23	00:09:46	01:09:56
	FConn_exp_1_gradient_sqp_fr_cd.xls	FConn_exp_1_gradient_slp_fr_cd.xls	FConn_exp_1_gradient_mmfd_fr_cd.xls	FConn_exp_1_gradient_suo_fr_cd.xls
	FConn_exp_1_gradient_sqp_fr_cd.tif	FConn_exp_1_gradient_slp_fr_cd.tif	FConn_exp_1_gradient_mmfd_fr_cd.tif	FConn_exp_1_gradient_suo_fr_cd.tif
	FConn_exp_1_gradient_sqp_fr_cd.out	FConn_exp_1_gradient_slp_fr_cd.out	FConn_exp_1_gradient_mmfd_fr_cd.out	FConn_exp_1_gradient_suo_fr_cd.out
	No feasible solution found	Maximum number of iterations reached	Relative objective convergence criterion met	Maximum number of iterations reached

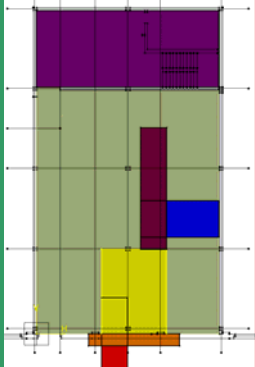
ANMERKUNG: Zeile K1-K54 der Datei: FConn\_exp\_1\_gradient\_suo\_fr\_cd.xls zeigt die Ergebnisse der VisualDOC-Berechnung. Die Daten wurden aus der folgenden Report-Datei von VisualDOC entnommen:  
FConn\_exp\_1\_gradient\_suo\_fr\_cd.out  
Das Excel-Interface zeigte andere Werte (B1-B54), die nicht mit den von VisualDOC errechneten Werten übereinstimmen. Auf Grund der richtigen Berechnungsergebnisse des Optimizers wurden dessen Werte als Grundlage der Bewertung in die Gesamtergebnisse übernommen.

Tab. 7.11 Force Connection FR

	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	FConn_exp1_gradient_sqp_suo_cd	FConn_exp1_gradient_slp_suo_cd	FConn_exp1_gradient_mmfd_suo_cd	FConn_exp1_gradient_suo_suo_cd
	gradient / sqp / suo / central difference	gradient / slp / suo / central difference	gradient / mmfd / suo / central difference	gradient / suo / suo / central difference
	nein	ja	ja	ja
	Kein Constraint erfüllt!	Alle Constraints vollständig erfüllt!	Alle Constraints vollständig erfüllt!	4 verletzte Constraints (max. 8 cm)
	Kein Flächen: 1,00 qm Kein Objective erfüllt!	3 Flächen 20,00 qm; 2 Flächen 19,99 qm Objectives vollständig erfüllt!	Alle Flächen 20,00 qm Objectives vollständig erfüllt!	F=20,03 qm Fh=20,02 qm Fg u. Fh=20,00 qm Fv=19,87 qm Objectives vollständig erfüllt!
	00:00:11	00:22:55	00:10:01	01:03:17
	FConn_exp_1_gradient_sqp_suo_cd.xls	FConn_exp_1_gradient_slp_suo_cd.xls	FConn_exp_1_gradient_mmfd_suo_cd.xls	FConn_exp_1_gradient_suo_suo_cd.xls
	FConn_exp_1_gradient_sqp_suo_cd.tif	FConn_exp_1_gradient_slp_suo_cd.tif	FConn_exp_1_gradient_mmfd_suo_cd.tif	FConn_exp_1_gradient_suo_suo_cd.tif
	FConn_exp_1_gradient_sqp_suo_cd.out	FConn_exp_1_gradient_slp_suo_cd.out	FConn_exp_1_gradient_mmfd_suo_cd.out	FConn_exp_1_gradient_suo_suo_cd.out
SUO Sequential Unconstrained Optimization (BIGDOT)	No feasible solution found	Maximum number of iterations reached	Relative objective convergence criterion met	Maximum number of iterations reached

ANMERKUNG: Zeile K1-K54 der Datei: FConn\_exp\_1\_gradient\_suo\_fr\_cd.xls zeigt die Ergebnisse der VisualDOC-Berechnung. Die Daten wurden aus der folgenden Report-Datei von VisualDOC entnommen:  
FConn\_exp\_1\_gradient\_suo\_suo\_cd.out  
Das Excel-Interface zeigte andere Werte (B1-B54), die nicht mit den von VisualDOC errechneten Werten übereinstimmen. Auf Grund der richtigen Berechnungsergebnisse des Optimizers wurden dessen Werte als Grundlage der Bewertung in die Gesamtergebnisse übernommen.

Tab. 7.12 Force Connection SUO

Force Connection		GA Genetic Algorithm
Task-Präfix:	FConn_	
Task-Nummer:	exp_1	
VisualDoc Task-Bezeichnung:	FConn_exp_1	FConn_exp1_non_gradient_ga
Task Type / Constrained Method / Unconstrained Method / Gradient		non-gradient / ga size:100 iterations:200
Task:	<p>Lower and Upper Bounds des 1. Constraints sind Null.                      Lower Bounds des 2. Constraints sind 1m, dies entspricht der Mindestbreite eines Verkehrsweges. Die Upper Bounds des 2. Constraints können gegen unendlich streben und könnten deshalb auf NONE gesetzt werden. Da der Genetic Algorithm jedoch bedingt, dass Integer-Werte verwendet werden, wurden die Upper Bounds auf 10 gesetzt. Da für den Berührungsbereich der jeweils kleinste Wert (MIN-Funktion) maßgebend ist, spielt der Wert des größeren Terms keine Rolle.                      Alle Inputs - mit Ausnahme der Konstanten (Type: Constant) wurden auf den Type: Integer gesetzt, da der GA dies verlangt.</p> <p>Overall Multiobjective Optimization Goal: Minimize</p> <p>Stopping Criteria:                      Relative Criteria Convergence: 1.00E-3                      Absolute Objective Convergence: 1.00E-6                      Max. Iterations: 200, Population Size: 100</p>	 <p>ANMERKUNG:                      Die Werte der Excel-Datei stimmen nicht mit den Werten aus der VisualDOC-Berechnung überein. Excel zeigt Werte an, nach denen die Objectives fast alle erfüllt sind. Die VisualDOC-Werte ergeben jedoch ein anderes Bild. Die Werte aus der Berechnung des Optimizers (VisualDOC) wurden manuell in der Excel-Tabelle eingetragen.</p>
Constraints:	<p><b>FConn</b>                      Umsetzung durch FConn_Constraint der Flächen (e, f, g, h, i).</p> <p><b>Force Connection (e, g) == (e north of g)</b>                      FConnN1 (e, g): <math>Y_{Ng} - Y_{Se} = 0</math>                      FConnN2 (e, g): <math>MIN(X_{Eg} - X_{We}, X_{Ee} - X_{Wg}) &gt;= 0</math></p> <p><b>Force Connection (f, g) == (f south of g)</b>                      FConnS1 (f, g): <math>Y_{Nf} - Y_{Sg} = 0</math>                      FConnS2 (f, g): <math>MIN(X_{Eg} - X_{Wf}, X_{Ef} - X_{Wg}) &gt;= 0</math></p> <p><b>Force Connection (h, g) == (h east of g)</b>                      FConnE1 (h, g): <math>X_{Eg} - X_{Wh} = 0</math>                      FConnE2 (h, g): <math>MIN(Y_{Ng} - Y_{Sh}, Y_{Nh} - Y_{Sg}) &gt;= 0</math></p> <p><b>Force Connection (i, g) == (i west of g)</b>                      FConnW1 (i, g): <math>X_{Ei} - X_{Wg} &gt;= 0</math>                      FConnW2 (i, g): <math>MIN(Y_{Ng} - Y_{Si}, Y_{Ni} - Y_{Sg}) &gt;= 0</math></p>	
Constraint-Erfüllung:		Relevant sind 2 veretzte Constraints East (h) und West (i).
Objectives: (VisualDOC Task Report)	Fe = 20qm, Ff = 20qm, Fg = 20qm, Fh = 20qm, Fi = 20qm	FF=20 qm, FF7 qm, Fg=20 qm, FF18 qm, FF12 qm Überschneidung erlaubt
Runtime:		00:59:24
VisualDoc-Excel-Output: File-Name		FConn_exp_1_non_gradient_ga.xls
Visualization:		FConn_exp_1_non_gradient_ga.tif
Summary-File:		FConn_exp_1_non_gradient_ga.out
Bemerkungen:		

Tab. 7.13 Force Connection Genetic Algorithm

## Beispiel 4

### Datei

siehe Tabelle (Tab. 7.15)

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.1)

### Zielfunktion

Target (Größe der Flächen  $F_f = 20\text{m}^2$ ,  $F_g = 20\text{m}^2$ ,  $F_h = 20\text{m}^2$ ,  $F_i = 20\text{m}^2$ )

### Nebenbedingungen

Als Nebenbedingung wird ausschließlich das Force Outside Constraint evaluiert.

### Berechnungszeit

siehe Tabelle (Tab. 7.15)

### Lösung

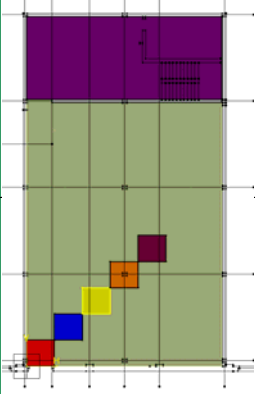
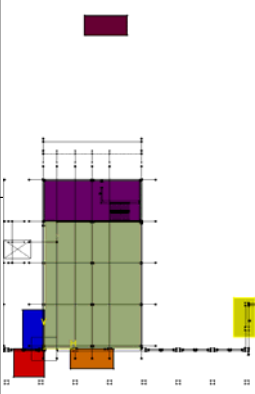
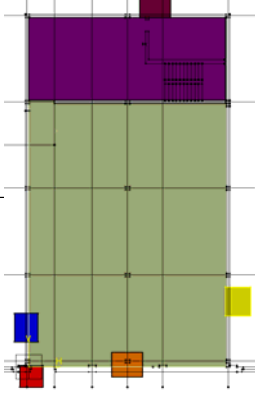
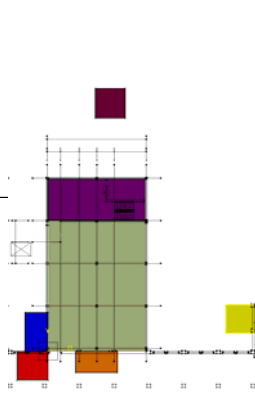
Das Force Outside Constraint wird verwendet, wenn eine Unit außerhalb der Grundunit  $j$  angeordnet werden muss. Diese Bedingung ist beispielsweise dann relevant, sobald die Gesamtfläche aller Units die Fläche der Grundunit übersteigt. In diesem Fall müssen einzelne Units außerhalb der Grundunit platziert werden. In der Grundrissplanung wird das Force Outside Constraint demgemäß als Anbau-Option eingesetzt. Das Force Outside Constraint gleicht im mathematischen Aufbau dem PInter Constraint. Da sich Units grundsätzlich nicht überschneiden dürfen, wird PInter für alle Units gleichermaßen festgelegt. Kommt es allerdings zu dem oben geschilderten Fall, dass Units außerhalb der Grundunit angeordnet werden müssen, kann dafür eine bestimmte Unit mit dem FOutside Constraint belegt werden. Durch dieses Constraint können zudem bestimmte Units sowohl mit PInter als auch mit FOutside belegt werden. PInter ist unabhängig von den Himmelsrichtungen, d.h. in keiner der vier Himmelsrichtungen darf es eine Überschneidung geben. Bei FOutside ist jedoch die Festlegung der Himmelsrichtung für die außerhalb der Grundunit anzuordnende Unit in vielen Fällen sinnvoll (z. B. Anbau eines Erschließungsbereiches im Norden der Grundrissfläche). Aus diesem Grund kann das Constraint in dieser Abhängigkeit definiert werden (z. B. FOutsideN).

Die nachfolgenden Tabellen verdeutlichen, dass lediglich die Algorithmen SLP und SUO alle Nebenbedingungen und Zielfunktionen vollständig erfüllen. Als Unconstrained-Methode wurde hier und bei allen nachfolgenden Constraints ausschließlich BFGS evaluiert, da die voran-

gegangenen Berechnungen keinerlei relevante Unterschiede von BFGS zu FR und SUO (BIGDOT) ergaben.

Force Outside		Globaler Index (Constraints)
Task-Präfix:	FOutside_	<b>Farbschemata:</b>
Task-Nummer:	exp_1	sehr gut
VisualDoc Task-Bezeichnung:	FOutside_exp_1	befriedigend
Task Type / Constrained Method / Unconstrained Method / Gradient		schlecht
Task:	<p>Flächen crimson, orange, gelb, blau, rot (e, f, g, h, i) außerhalb der Grundriss (j) anordnen. Die Flächen (e, f, g, h) werden in Abhängigkeit von den Himmelsrichtungen nach N / S / E / W ausgerichtet. Die Fläche (i) wird unabhängig, d.h. ohne vorherige Definition der Ausrichtung angeordnet. Die Flächen dürfen sich überschneiden. Alle anderen Constraints sind deaktiviert. Ausschließlich FOutside_exp_1 wird evaluiert.</p> <p>Overall Multiobjective Optimization Goal: Minimize</p> <p>Stopping Criteria:                      Relative Objective Convergence: 1.00E-3                      Absolute Objective Convergence: 1.00E-6</p> <p>Max. Iterations: 100, Iterations for Convergence: 10</p>	<p><b>Akronyme: Constrained Method</b>                      SQP - Sequential Quadratic Programming                      SLP - Sequential Linear Programming                      MMFD - Modified Method of Feasible Directions                      SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Unconstrained Method</b>                      BFGS - Broydon-Fletcher-Goldfarb-Shanno                      FR - Fletcher-Reeves                      SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Gradients</b>                      CD - Central Difference                      FD - Forward Difference</p>
Constraints:	<p><b>FOutside</b>                      Umsetzung durch FOutside_Constraint der Flächen (e, f, g, h).  <b>Force Outside (e, j) == (e north of j)</b>                      FOutsideN(e, j, Y_Nj - Y_Se) &lt;= 0  <b>Force Outside (f, j) == (f south of j)</b>                      FOutsideS(f, j, Y_Nf - Y_Sj) &lt;= 0  <b>Force Outside (g, j) == (g east of j)</b>                      FOutsideE(g, j, X_Eg - X_Wg) &lt;= 0  <b>Force Outside (h, j) == (h west of j)</b>                      FOutsideW(h, j, X_Eh - X_Wh) &lt;= 0  <b>FOutside</b>                      Umsetzung durch FOutside_Constraint der Fläche (i).  <b>Force Outside (i, j) == (i north / south / east / west of j)</b>                      MIN(Y_Ni - Y_Si : Y_Ni - Y_Si, X_Ei - X_Wi, X_Ei - X_Wi) &lt;= 0</p>	
Constraint-Erfüllung:		
Objectives: (VisualDOC Task Report)	Ff = 20qm, Fg = 20qm, Fh = 20qm, Fi = 20qm	
Runtime:		
VisualDoc-Excel-Output: File-Name		
Visualization:		
Summary-File:		
Bemerkungen:		

Tab. 7.14 Force Outside

	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	FOutside_exp1_gradient_sqp_bfqs_cd	FOutside_exp1_gradient_slp_bfqs_cd	FOutside_exp1_gradient_mmfd_bfqs_cd	FOutside_exp1_gradient_suo_bfqs_cd
	gradient / sqp / bfqs / central difference	gradient / slp / bfqs / central difference	gradient / mmfd / bfqs / central difference	gradient / suo / bfqs / central difference
	nein	ja	nein	ja
				
	Kein Constraint erfüllt !	Alle Constraints vollständig erfüllt !	3 verletzte Constraints (h: 0.96m; (h) 0.81m; (l) 1.02m)	Alle Constraints vollständig erfüllt !
	1 Fläche 4.03 qm Kein Objective erfüllt !	3 Flächen 20.00 qm; 2 Flächen 18.99 qm Objectives vollständig erfüllt !	1 Fläche 20.00 qm; 1 Fläche 18.99 qm; 1 Fläche 18.99 qm Kein Objective erfüllt !	1 Fläche 20.00 qm; 4 Flächen 18.99 qm Objectives vollständig erfüllt !
	00:00:16	00:40:57	00:03:04	nach 03:55:24 abgebrochen (tatsächliche Zeit: 00:39:10)
	FOutside_exp_1_gradient_sqp_bfqs_cd.xls	FOutside_exp_1_gradient_slp_bfqs_cd.xls	FOutside_exp_1_gradient_mmfd_bfqs_cd.xls	FOutside_exp_1_gradient_suo_bfqs_cd.xls
	FOutside_exp_1_gradient_sqp_bfqs_cd.tif	FOutside_exp_1_gradient_slp_bfqs_cd.tif	FOutside_exp_1_gradient_mmfd_bfqs_cd.tif	FOutside_exp_1_gradient_suo_bfqs_cd.tif
	FOutside_exp_1_gradient_sqp_bfqs_cd.out	FOutside_exp_1_gradient_slp_bfqs_cd.out	FOutside_exp_1_gradient_mmfd_bfqs_cd.out	FOutside_exp_1_gradient_suo_bfqs_cd.out
	No feasible solution found	Maximum number of iterations reached	No feasible solution found  Im Ergebnis ist erkennbar, dass der Optimizer die Constraints ansatzweise erfüllt. Aus diesem Grund wurden zwei weitere Optimierungsdurchgänge mit dem jeweils "Best Design Point" als Startpunkt durchgeführt. Der letzte Durchgang führte zu einem Ergebnis, bei welchem alle Constraints vollständig erfüllt wurden. Die Objectives ändern sich jedoch nicht. Sie bleiben weiterhin mit fast unveränderten Werten unerfüllt.	Task aborted!!  Der Task wurde händisch abgebrochen, da die Berechnungszeit bei knapp 4 Stunden lag. Die Report-Datei von VisualDOC zeigt jedoch die folgenden Werte an:  * Run Started On : Tue Dec 07 16:46:58 2004 * Run Stopped On : Tue Dec 07 17:26:08 2004  Aus diesen Werte geht hervor, dass der Task bereits vorher regulär beendet worden ist, ohne dass das Programm in den Ursprungszustand zurückgesetzt wurde. Aus der Report-Datei wurden die Excel-Werte manuell übernommen und evaluiert.

BFGS  
Broyden-Fletcher-Goldfarb-Shanno

Tab. 7.15 Force Outside BFGS

## Beispiel 5

### Datei

siehe Tabelle (Tab. 7.17)

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.1)

### Zielfunktion

Target (Größe der Flächen  $F_f = 20\text{m}^2$ ,  $F_g = 20\text{m}^2$ ,  $F_h = 20\text{m}^2$ ,  $F_i = 20\text{m}^2$ )

### Nebenbedingungen

Als Nebenbedingung wird ausschließlich das Force To Border Constraint evaluiert.

### Berechnungszeit

siehe Tabelle (Tab. 7.17)

### Lösung

Die nachfolgenden Tabellen verdeutlichen, dass der SQP-Algorithmus keine der Nebenbedingungen und Zielfunktionen erfüllt, während SLP und SUO bereits im ersten Berechnungsdurchgang alle Nebenbedingungen und Zielfunktionen vollständig erfüllen. Da der MMFD-Algorithmus gegen Ende der Berechnung eine Erfüllung der Constraints anzeigt, wurde eine erneute Berechnung unter Verwendung des Best-Design-Point, d.h. unter Verwendung des besten bisher erzielten Ergebnisses, durchgeführt. Diese Best-Design-Point Berechnung führt daraufhin zur Erfüllung aller Nebenbedingungen und Zielfunktionen. Die Anwendung dieser Vorgehensweise ist dann empfehlenswert, wenn bei komplexen Berechnungen im ersten Durchgang kein globales Optimum gefunden werden kann, jedoch eine Annäherung an eine potentielle Lösung des Problems während der Berechnung erfolgt ist.



Force To Border		Globaler Index (Constraints)	
Task-Präfix:	F2Border_	<b>Farbschemata:</b>	
Task-Nummer:	exp_1	sehr gut	
VisualDoc Task-Bezeichnung:	F2Border_exp_1	befriedigend	
Task Type / Constrained Method / Unconstrained Method / Gradient		schlecht	
Task:	<p>Flächen crimson, orange, gelb, blau, rot (e, f, g, h, i) auf der inneren Begrenzungslinie der Grundunit (j) anordnen. Die Flächen (e, f, g, h) werden in Abhängigkeit von den Himmelsrichtungen nach N / S / E / W ausgerichtet. Die Fläche (i) wird unabhängig, d.h. ohne vorherige Definition der Ausrichtung angeordnet. Die Flächen dürfen sich überschneiden. Alle anderen Constraints sind deaktiviert. Ausschließlich F2Border_exp_1 wird evaluiert.</p> <p>Overall Multiobjective Optimization Goal: Minimize</p> <p>Stopping Criteria:                      Relative Objective Convergence: 1.00E-3                      Absolute Objective Convergence: 1.00E-6</p> <p>Max. Iterations: 100, Iterations for Convergence: 10</p>	<p><b>Akronyme: Constrained Method</b>                      SQP - Sequential Quadratic Programming                      SLP - Sequential Linear Programming                      MMFD - Modified Method of Feasible Directions                      SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Unconstrained Method</b>                      BFGS - Broydon-Fletcher-Goldfarb-Shanno                      FR - Fletcher-Reeves                      SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Gradients</b>                      CD - Central Difference                      FD - Forward Difference</p>	
Constraints:	<p><b>F2Border</b>                      Umsetzung durch F2Border_Constraint der Flächen (e, f, g, h).  <b>Force To Border (e, j) == (e inside northern border of j)</b>  <math>F2BorderN(e, j, Y_{Ne} - Y_{Nj}) = 0</math>  <b>Force To Border (f, j) == (f inside southern border of j)</b>  <math>F2BorderS(f, j, Y_{Sf} - Y_{Sj}) = 0</math>  <b>Force To Border (g, j) == (g inside eastern border of j)</b>  <math>F2BorderE(g, j, X_{Eg} - X_{Ej}) = 0</math>  <b>Force To Border (h, j) == (h inside western border of j)</b>  <math>F2BorderW(h, j, X_{Wg} - X_{Wj}) = 0</math>  <b>FOutside</b>                      Umsetzung durch F2Border_Constraint der Fläche (i).  <b>Force To Border (i, j) == (i inside northern / southern / eastern / western border of j)</b>  <math>MIN(Y_{Ni} - Y_{Ni}, Y_{Si} - Y_{Si}, X_{Ei} - X_{Ej}, X_{Wi} - X_{Wj}) = 0</math></p>		
Constraint-Erfüllung:			
Objectives: (VisualDOC Task Report)	Ff = 20qm, Fg = 20qm, Fh = 20qm, Fi = 20qm		
Runtime:			
VisualDoc-Excel-Output: File-Name			
Visualization:			
Summary-File:			
Bemerkungen:			
Task-Präfix:	F2Border_		
Task-Nummer:	exp_2		
VisualDoc Task-Bezeichnung:	F2Border_exp_2		
Task Type / Constrained Method / Unconstrained Method / Gradient			
Constraint-Erfüllung:			
Objectives: (VisualDOC Task Report)			
Runtime:			
VisualDoc-Excel-Output: File-Name			
Visualization:			
Summary-File:			

Tab. 7.16 Force To Border

	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	F2Border_exp1_gradient_sqp_bfgs_cd	F2Border_exp1_gradient_slp_bfgs_cd	F2Border_exp1_gradient_mmfd_bfgs_cd	F2Border_exp1_gradient_suo_bfgs_cd
	gradient / sqp / bfgs / central difference	gradient / slp / bfgs / central difference	gradient / mmfd / bfgs / central difference	gradient / suo / bfgs / central difference
	nein	ja	ja	nein
	Ein Constraint erfüllt, welches bereits in der Ausgangsposition erfüllt war!	Alle Constraints vollständig erfüllt!	Alle Constraints vollständig erfüllt!	Alle Constraints vollständig erfüllt!
	Alle Flächen 20,00 qm Kein Objective erfüllt!	1 Fläche 20,00 qm, zwei Flächen 19,99 qm, zwei Flächen 19,97 qm. Objectives vollständig erfüllt!	Alle Flächen 20,00 qm Kein Objective erfüllt!	Alle Flächen 20,00 qm Objectives vollständig erfüllt!
	00:00:15	00:08:59	00:03:32	01:47:06
	F2Border_exp_1_gradient_sqp_bfgs_cd.xls	F2Border_exp_1_gradient_slp_bfgs_cd.xls	F2Border_exp_1_gradient_mmfd_bfgs_cd.xls	F2Border_exp_1_gradient_suo_bfgs_cd.xls
	F2Border_exp_1_gradient_sqp_bfgs_cd.tif	F2Border_exp_1_gradient_slp_bfgs_cd.tif	F2Border_exp_1_gradient_mmfd_bfgs_cd.tif	F2Border_exp_1_gradient_suo_bfgs_cd.tif
	F2Border_exp_1_gradient_sqp_bfgs_cd.out	F2Border_exp_1_gradient_slp_bfgs_cd.out	F2Border_exp_1_gradient_mmfd_bfgs_cd.out	F2Border_exp_1_gradient_suo_bfgs_cd.out
	No feasible solution found	Violated constraint with zero gradient	Kuhn-Tucker conditions met Der Graph der Zielfunktion zeigt eine Erfüllung der Constraints gegen Ende der Berechnung. Aus diesem Grund wurde die Optimierungsrechnung mit dem Best-Design-Point erneut durchgeführt. Das Ergebnis dieser Berechnung weist eine Erfüllung aller Constraints und Objectives auf.	Task aborted! 1. Anm.: Der Task wurde händisch abgebrochen, da die Berechnung offensichtlich beendet war. Die Report-Datei von VisualDOC bestätigt dies und zeigt die folgenden Werte an: • Run Started On : Fri Dec 10 19:57:51 2004 • Run Stopped On : Fri Dec 10 21:44:57 2004 Aus diesen Werte geht hervor, dass der Task bereits vorher regulär beendet worden ist, ohne dass das Programm in den Ursprungszustand zurückgesetzt wurde. Aus der Report-Datei wurden die Excel-Werte manuell übernommen und evaluiert. 2. Anm.: Aus dem Berechnungsergebnis ist abzulesen, dass alle Constraints erfüllt sind. Dennoch ist die Fläche i (rot) ausserhalb der Grundunit angeordnet. Das F2Border-Constraint ist derart definiert, dass davon ausgegangen wird, dass ein anderes Constraint (Finside) die Aufgabe übernimmt alle Units innerhalb der Grundfläche j anzuordnen. (siehe bemerkung.doc)
			F2Border_exp2_gradient_mmfd_bfgs_cd	
			gradient / mmfd / bfgs / central difference	
			Alle Constraints vollständig erfüllt!	
			Alle Flächen 20,00 qm Objectives vollständig erfüllt!	
			00:02:01	
			F2Border_exp_2_gradient_mmfd_bfgs_cd.xls	
			F2Border_exp_2_gradient_mmfd_bfgs_cd.tif	
			F2Border_exp_2_gradient_mmfd_bfgs_cd.out	

BFGS  
Brüder-Flacher-Goldfisch-Shampoo

Tab. 7.17 Force To Border BFGS

## Beispiel 6

### Datei

siehe Tabelle (Tab. 7.19)

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.1)

### Zielfunktion

Target (Größe der Flächen  $F_f = 20\text{m}^2$ ,  $F_g = 20\text{m}^2$ ,  $F_h = 20\text{m}^2$ ,  $F_i = 20\text{m}^2$ )

### Nebenbedingungen

Als Nebenbedingung wird ausschließlich das Prohibit Connection Constraint evaluiert.

### Berechnungszeit

siehe Tabelle (Tab. 7.19)

### Lösung

Die nachfolgenden Abbildungen verdeutlichen, dass der SQP-Algorithmus keine der Nebenbedingungen und Zielfunktionen erfüllt, während SLP und MMFD alle Nebenbedingungen und Zielfunktionen vollständig erfüllen. SUO erfüllt die Nebenbedingungen und Zielfunktionen nur annähernd.

Prohibit Connection		Globaler Index (Constraints)	
Task-Präfix:	PConn_	<b>Farbschemata:</b>	
Task-Nummer:	exp_1	sehr gut	
VisualDoc Task-Bezeichnung:	PConn_exp_1	befriedigend	
Task Type / Constrained Method / Unconstrained Method / Gradient		schlecht	
Task:	<p>Flächen crimson, orange, blau, rot (e, f, h, i) ohne Verbindung zur Fläche gelb (g) anordnen. Keine der Flächen darf die Fläche gelb (g) berühren. Das Mindestmaß für den Abstand zur Fläche gelb (g) wird durch Upper- und Lower Bounds festgelegt. Dieses Mindestmaß muß größer als Null sein. Die Flächen müssen nicht innerhalb der Grundriss belege (j) liegen. Die Flächen dürfen sich überschneiden. Alle anderen Constraints sind deaktiviert. Ausschließlich PConn_exp_1 wird evaluiert.</p> <p>Lower und Upper Bounds des PConn-Constraints müssen kleiner als Null sein. Beide Bounds werden auf -1 gesetzt, dies entspricht der Mindestbreite (1m) eines Verkehrsweges.</p> <p>Overall Multiobjective Optimization Goal: Minimize</p> <p>Stopping Criteria:                      Relative Objective Convergence: 1.00E-3                      Absolute Objective Convergence: 1.00E-6</p> <p>Max. Iterations: 100, Iterations for Convergence: 10</p>	<p><b>Akronyme: Constrained Method</b>                      SQP - Sequential Quadratic Programming                      SLP - Sequential Linear Programming                      MMFD - Modified Method of Feasible Directions                      SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Unconstrained Method</b>                      BFGS - Broydon-Fletcher-Goldfarb-Shanno                      FR - Fletcher-Reeves                      SUO - Sequential Unconstrained Optimization (BIGDOT)</p> <p><b>Akronyme: Gradients</b>                      CD - Central Difference                      FD - Forward Difference</p>	
Constraints:	<p><b>PConn</b>                      Umsetzung durch PConn_Constraint der Flächen (e, f, h, i).  <b>Prohibit Connection (e, g) == (e does not connect to g)</b>                      PConn (e, g):  <math>MIN(Y_{Ng} - Y_{Se}, Y_{Ne} - Y_{Sg}, X_{Eg} - X_{We}, X_{Ee} - X_{Wg}) &lt; 0</math>  <b>Prohibit Connection (f, g) == (f does not connect to g)</b>                      PConn (f, g):  <math>MIN(Y_{Ng} - Y_{Sf}, Y_{Nf} - Y_{Sg}, X_{Eg} - X_{Wf}, X_{Ef} - X_{Wg}) &lt; 0</math>  <b>Prohibit Connection (h, g) == (h does not connect to g)</b>                      PConn (h, g):  <math>MIN(Y_{Ng} - Y_{Sh}, Y_{Nh} - Y_{Sg}, X_{Eg} - X_{Wh}, X_{Eh} - X_{Wg}) &lt; 0</math>  <b>Prohibit Connection (i, g) == (i does not connect to g)</b>                      PConn (i, g):  <math>MIN(Y_{Ng} - Y_{Si}, Y_{Ni} - Y_{Sg}, X_{Eg} - X_{Wi}, X_{Ei} - X_{Wg}) &lt; 0</math></p>		
Constraint-Erfüllung:			
Objectives: (VisualDOC Task Report)	Pf = 20qm, Fg = 20qm, Fh = 20qm, Fi = 20qm		
Runtime:			
VisualDoc-Excel-Output: File-Name			
Visualizaton:			
Summary-File:			
Bemerkungen:			

Tab. 7.18 Prohibit Connection

	SQP Sequential Quadratic Programming	SLP Sequential Linear Programming	MMFD Modified Method of Feasible Directions	SUO Sequential Unconstrained Optimization (BIGDOT)
	PConn_exp1_gradient_sqp_bfgs_cd	PConn_exp1_gradient_slp_bfgs_cd	PConn_exp1_gradient_mmfd_bfgs_cd	PConn_exp1_gradient_suo_bfgs_cd
	gradient / sqp / bfgs / central difference	gradient / slp / bfgs / central difference	gradient / mmfd / bfgs / central difference	gradient / suo / bfgs / central difference
	nein	ja	ja	ja
	Kein Constraint erfüllt!	Alle Constraints vollständig erfüllt!	Alle Constraints vollständig erfüllt!	Alle Constraints vollständig erfüllt! (Verletzte Constraints liegen im Bereich von 0.02 oder kleiner)
	Kein Flächen 1.00 qm Kein Objective erfüllt!	3 Flächen 19.99 qm; 2 Flächen 20.00 qm Objectives vollständig erfüllt!	4 Flächen 20.00 qm; 1 Flächen 19.99 qm Objectives vollständig erfüllt!	Fläche 1.00 qm; Fläche 2.00 qm; Fläche 3.00 qm; Fläche 4.00 qm F=19.94 qm, Objectives annähernd erfüllt!
	00:00:19	00:30:09	00:01:43 !!!	00:50:13
	PConn_exp_1_gradient_sqp_bfgs_cd.xls	PConn_exp_1_gradient_slp_bfgs_cd.xls	PConn_exp_1_gradient_mmfd_bfgs_cd.xls	PConn_exp_1_gradient_suo_bfgs_cd.xls
	PConn_exp_1_gradient_sqp_bfgs_cd.tif	PConn_exp_1_gradient_slp_bfgs_cd.tif	PConn_exp_1_gradient_mmfd_bfgs_cd.tif	PConn_exp_1_gradient_suo_bfgs_cd.tif
	PConn_exp_1_gradient_sqp_bfgs_cd.out	PConn_exp_1_gradient_slp_bfgs_cd.out	PConn_exp_1_gradient_mmfd_bfgs_cd.out	PConn_exp_1_gradient_suo_bfgs_cd.out
	No feasible solution found	Maximum number of iterations reached Die Flächen f und i befinden sich hinter den Flächen e und h. Der Abstand zur Fläche g beträgt in x- und x-Richtung jeweils 1 m.	Relative objective convergence criterion met Die Flächen f und i befinden sich hinter den Flächen e und h. Der Abstand zur Fläche g beträgt in x- und x-Richtung jeweils 1 m.	Task aborted!!! Stop Code: maximum iterations reached  1. Anm.: Der Task wurde händisch abgebrochen, da die Berechnung offensichtlich beendet war. Die Report-Datei von VisualDOC bestätigt dies und zeigt die folgenden Werte an:  * Run Started On : Sun Dec 12 23:10:00 2004 * Run Stopped On : Mon Dec 13 00:00:13 2004  Aus diesen Werte geht hervor, dass der Task bereits vorher regulär beendet worden ist, ohne dass das Programm in den Ursprungsstand zurückgesetzt wurde. Aus der Report-Datei wurden die Excel-Werte manuell übernommen und evaluiert.  Die Flächen f und h befinden sich hinter der Fläche e. Der Abstand zur Fläche g beträgt in x- und x-Richtung jeweils etwa 1 m.

BFGS  
Broyden-Fletcher-Goldfarb-Shanno

Tab. 7.19 Prohibit Connection BFGS

### 7.2.4 Bewertung

Die Beispiele 1-6 verdeutlichen die grundsätzliche Funktionalität des Prototyps. Mit Hilfe der verwendeten Constrained Methoden (SLP, MMFD und SUO) ist es möglich, die definierten Nebenbedingungen und Zielfunktionen zu erfüllen. Die Wahl der Unconstrained Methode (BFGS, FR, SUO) hat keinen Einfluss auf das Ergebnis einer Berechnung. Dieses Verhalten war vorhersehbar, da in den Optimierungsproblemen Constraints verwendet werden. Die Evaluation wurde jedoch durchgeführt, weil nicht dokumentiert ist, ob die zusätzliche Verwendung der Unconstrained Methode Auswirkungen auf die nicht durch Nebenbedingungen beschränkten Variablen eines Optimierungsproblems hat. Als stabilstes Verfahren mit den besten Ergebnissen ist SLP zu werten. SUO liefert ebenfalls gute Ergebnisse, ist jedoch mit extrem hohen Berechnungskosten verbunden. MMFD benötigt geringe Berechnungskosten, liefert jedoch fehlerhafte Ergebnisse. Die Ergebnisse der SQP-Methode sind sowohl bei linearen als auch bei quadratischen Zielfunktionen unbrauchbar.

Im Verlauf weiterer Berechnungen wurde versucht, den Prototyp zur Lösung komplexerer Probleme einzusetzen. Die Ergebnisse sind ernüchternd. Keines der definierten komplexen Probleme konnte zufrieden stellend gelöst werden. Während die Anwendung einzelner Constraints optimale Lösungen hervorbringt, führt die Verwendung von Constraint-Kombinationen zu Ergebnissen mit verletzen Constraints und nicht erfüllten Zielfunktionen. In VisualDOC tritt eine Constraintverletzung auf, wenn der Constraint-Wert positiver ist als der Wert der Variable CTMIN (Constraint Tolerance). Der Default-Wert für CTMIN beträgt 0.003. Dadurch wird die Genauigkeit für die Erfüllung der Constraints definiert. Da der Default-Wert eine sehr hohe Genauigkeit darstellt, wurde mit verminderten Werten experimentiert. Bei einer Berechnung mit CTMIN 0.003 wurde kein Ergebnis gefunden (No Feasible Solution Found). Eine Erhöhung des Wertes auf 0.1 führte zu einer Erfüllung aller Constraints. Die Zielfunktion wurde nur teilweise erfüllt. Ein zweiter Berechnungsdurchgang mit dem Best Design Point lieferte ebenfalls nur eine teilweise erfüllte Zielfunktion. Eine erneute Erhöhung von CTMIN auf 1.00 erbrachte eine Erfüllung aller Constraints unter Beachtung des neuen CTMIN-Wertes (d.h. die Units lagen im Extremfall knapp 1m auseinander bzw. überschnitten sich um bis zu 1m). Die Zielfunktion wurde vollständig erfüllt (Abb. 7.5).

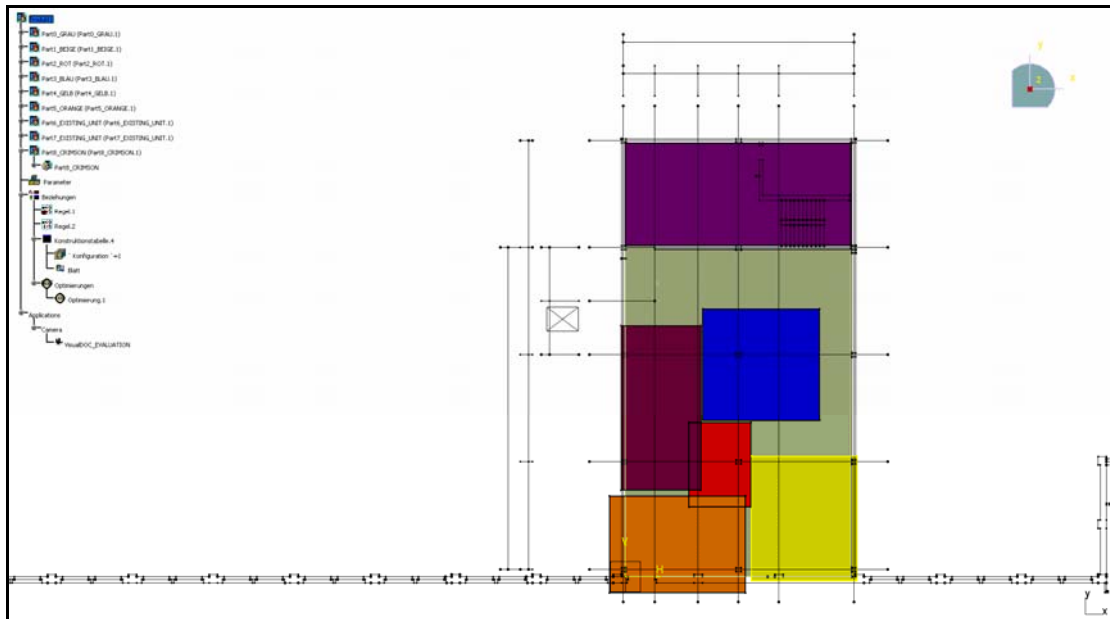


Abb. 7.5 CTMIN 1 - Constraint Erfüllung im Toleranzbereich 1 und vollständige Erfüllung der Zielfunktion

Die Berechnungsergebnisse der Design-Variablen wurden daraufhin in eine neue Berechnung mit Default-CTMIN 0.003 eingefügt. Die Ausgangskonfiguration für die Anordnung der Units entsprach somit einer wünschenswerten Start-Topologie. Das Berechnungsergebnis zeigt eine vollständige Erfüllung aller Constraints und der Zielfunktion. (Abb. 7.6)

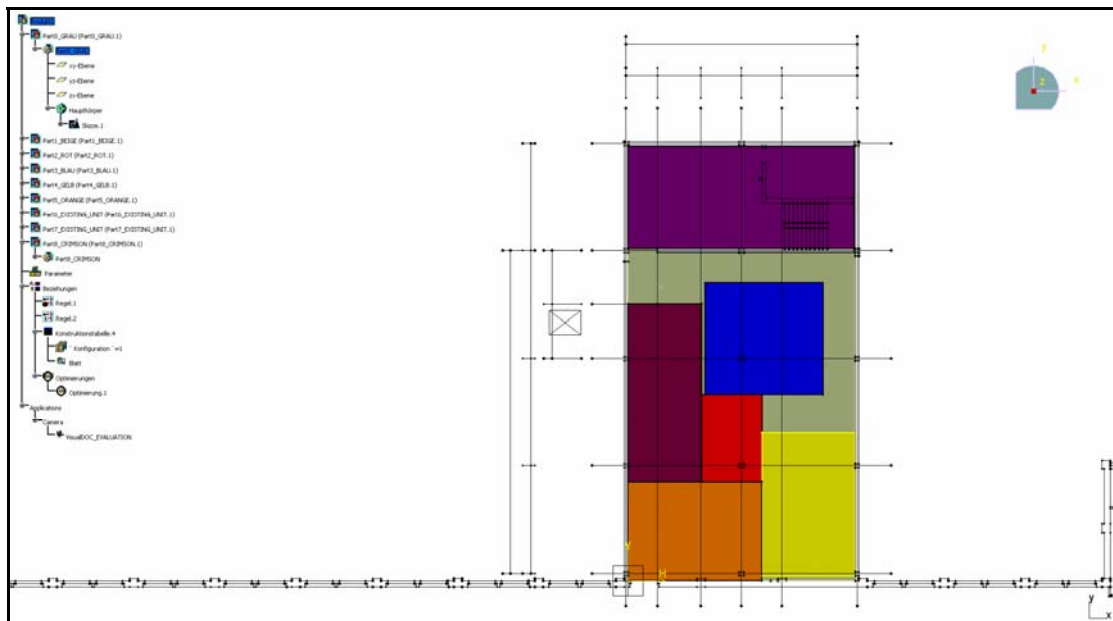


Abb. 7.6 CTMIN 0.003 - Basierend auf der Berechnung mit dem Best-Design-Point von CTMIN 1

Die Ergebnisse belegen, dass Anordnungsprobleme, für welche keine zulässige Lösung gefunden werden kann, mit einer höheren Constraint-Toleranz zu lösen sind. Durch diese Vorgehensweise werden die Units in eine Position gebracht, die einer guten Start-Topologie entspricht. Die Verringerung der Constraint-Toleranz kann, in Kombination mit einer erneuten Berechnung unter Verwendung der Best-Design-Points der ursprünglichen Berechnung, zu einer Erfüllung aller Constraints und der Zielfunktion führen. Abb. 7.7 zeigt ein Ergebnis dieser Vorgehensweise. Die FConn- und PInter- Constraints lieferten im ersten Berechnungsdurchgang die gewünschten Ergebnisse. Für das FInside Constraint konnte erst im dritten Berechnungsdurchgang, der jeweils mit dem Best Design Point aus der vorangegangenen Berechnung durchgeführt wurde, ein Ergebnis erzielt werden, bei dem alle Constraints und die Zielfunktion vollständig und gleichzeitig erfüllt wurden. Die ersten zwei Berechnungen wurden vom Optimizer - mit dem Ergebnis „No Feasible Solution Found“ - abgebrochen. Die Berechnungszeit lag bei allen Berechnungen bei wenigen Minuten.



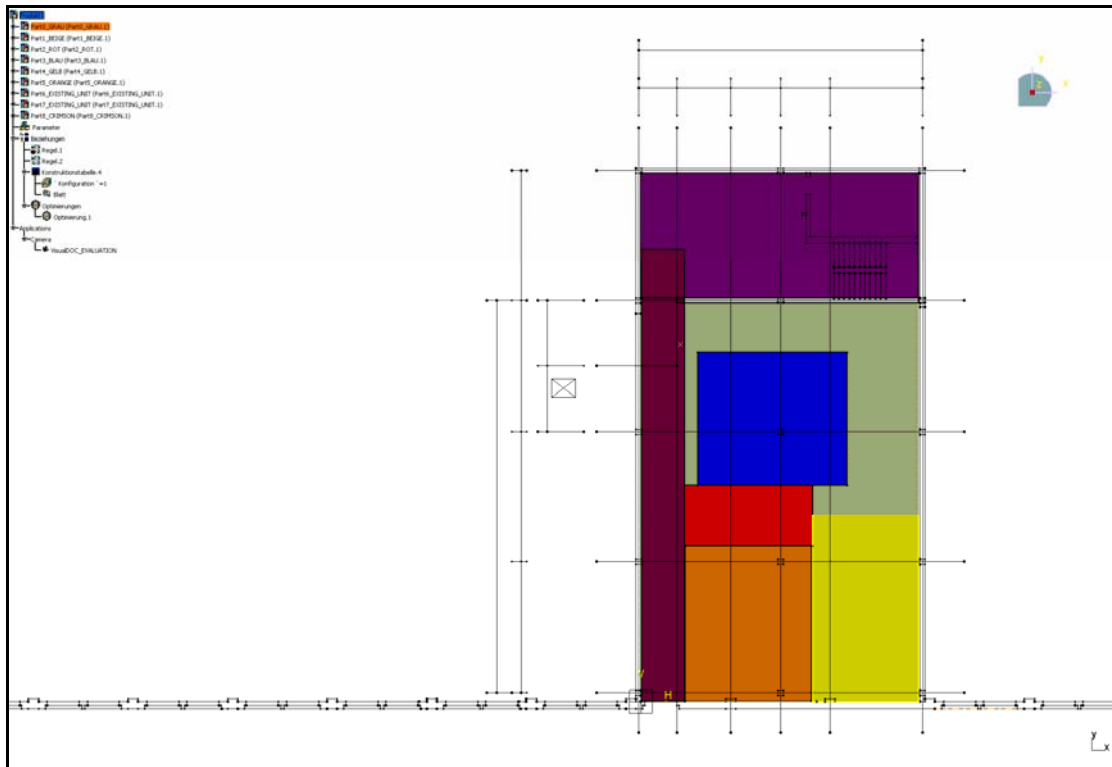


Abb. 7.7 SLP-Berechnung - Erfüllung aller Constraints und der Target-Zielfunktion

Um den Einfluss eines fehlerhaften Excel-Interfaces auszuschließen, wurde im Verlauf weiterer Berechnungen das C++ API zur Evaluation des Prototyps verwendet. Die Qualität der Ergebnisse war jedoch, mit denen des Excel-Interfaces, identisch. Zusammenfassend lässt sich festhalten, dass der VisualDOC Prototyp die Funktionsfähigkeit des entwickelten geometrischen und mathematischen Modells (vgl. Kap. 5.2.1 und Kap. 6.1.1) beweist. Die Deklaration einer Unit mit 4 Abstandsvariablen beinhaltet die theoretische Möglichkeit, dass die gleichzeitige Änderung der Abszisse und Ordinate des Referenzpunktes und der Abstandsvariablen trotzdem zu keiner Änderung der Größe oder Position einer Unit führen kann. Dieses Verhalten wurde bei dem Prototyp jedoch nicht beobachtet. Zur Verringerung der Variablenanzahl wurden bei den nachfolgenden Prototypen mathematische Modelle verwendet, die auf 2 Abstandsvariablen basieren (vgl. Kap. 6.1.2 und Kap. 6.1.3). Zur Lösung komplexerer Probleme mit mehreren Nebenbedingungen ist der Prototyp VisualDOC - auch auf Grund des Fehlens benutzerdefinierter Suchverfahren - ungeeignet. Es ist nur bedingt möglich, durch die gezielte Festlegung geeigneter Start-Topologien und mehrfacher Berechnung des Optimierungsproblems unter Verwendung der jeweiligen Best-Design-Points zu einer optimalen Lösung zu gelangen. Diese Vorgehensweise ist zur Lösung von Planungsaufgaben aus dem Bereich der Zerstörungsreichen Modellbildung zu aufwändig und nicht praktikabel. Derartige Strategien, die Start-Topologien benutzen, gelangen im Operations Research jedoch durchaus zur Anwendung und werden in der Literatur auch explizit als Eröffnungsverfahren beschrieben [Domschke und Drexl, 2005].

## 7.3 OPL

### 7.3.1 Konzeptioneller Aufbau

Evaluiertes Einsatzbereich ZR + ZA

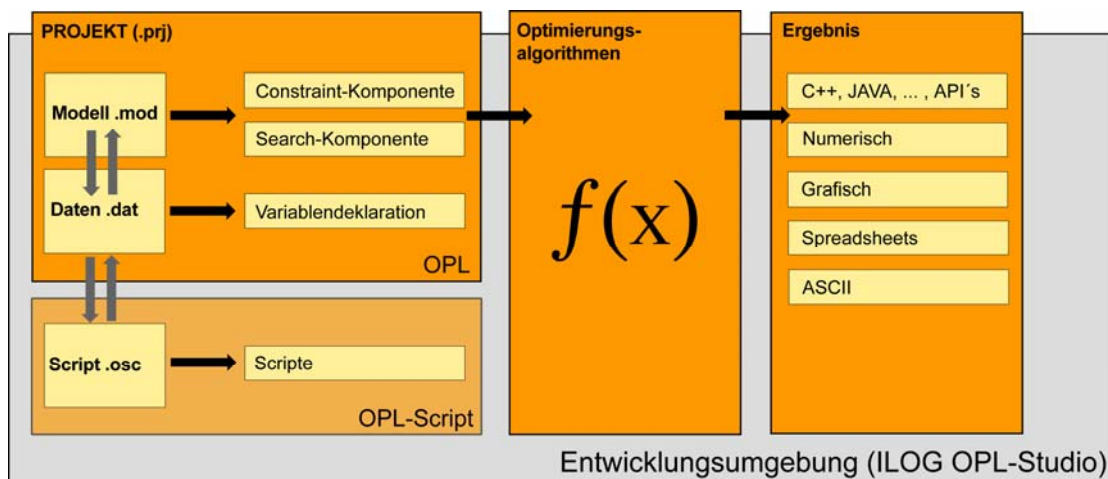


Abb. 7.8 Konzeptioneller Aufbau OPL-Prototyp

Die Implementierung des Prototyps wurde mit der kommerziellen Entwicklungsumgebung ILOG OPL Studio (Version 3.7 und 3.7.1) durchgeführt. OPL Studio ist eine integrierte graphische Entwicklungsumgebung für die Mathematische Programmierung und die Darstellung und Lösung kombinatorischer Optimierungsprobleme. Sie ist die einzige Umgebung innerhalb derer die Modellierungssprache OPL (Optimization Programming Language) angewendet werden kann. In OPL Studio können Programme in OPL geschrieben und ausgeführt werden. Das Produkt erlaubt die Kompilierung des erzeugten Programmcodes und dessen Einbindung in Sprachen wie C++ und Java. Es erlaubt ebenfalls die Einbindung der Solver in externen Programmcodes. Neben OPL wurde bei der Erstellung des Prototyps auch die Scriptsprache OPL-Script verwendet. OPL-Script interagiert mit OPL, um bspw. Instanzen eines Problems zu lösen und die Ergebnisse an OPL weiterzugeben. Der prinzipielle Aufbau eines Optimierungsproblems in OPL wird mit vier unterschiedlichen Datentypen realisiert:

**Modelle (.mod)**

Diese beinhalten den eigentlichen OPL-Programmcode und können als Stand-Alone Modelle in OPL Studio ausgeführt werden. Modellfiles beinhalten bei Problemen geringer Komplexität auch Daten.

**Daten (.dat)**

Datenfiles werden verwendet, um problemspezifische Daten vom Programmcode zu trennen. Diese Vorgehensweise ist bei komplexen Problemen sowie bei Modellen empfehlenswert, auf welche unterschiedliche Datensätze angewendet werden sollen.

**Projekte (.prj)**

Um Modellfiles mit Datenfiles zu verbinden, werden diese in Projektfiles zusammengeführt. Das Modellfile deklariert die Daten, initialisiert sie jedoch nicht. Die Initialisierung wird durch das Datenfile bewerkstelligt.

**Scripte (.osc)**

Diese beinhalten OPL-Scriptcode und assoziieren Modell- und Datenfiles mit dem Script.

### 7.3.2 Besonderheiten

Die Besonderheiten dieses Prototyps liegen in der Verwendung der Modellierungssprache OPL (Optimization Programming Language). Die Sprache wurde 1995 von Pascal Van Hentenryck auf der Basis von Modellierungssprachen wie AMPL und GAMS entwickelt [Van Hentenryck und Lustig, 1999]. Diese Sprachen verwenden eine Syntax auf der Basis algebraischer Notation, welche in mathematisch-wissenschaftlichen Veröffentlichungen allgemein Verwendung findet. Ihr Haupteinsatzbereich liegt in der kombinatorischen Optimierung, insbesondere in der Lösung von Problemen der Mathematischen Programmierung. Programmierung sei in diesem Fall nicht als Programmiersprache im Sinn der Informatik, sondern als eine Klassifikation eines determinierten mathematischen Problembereichs, verstanden. Zur Mathematischen Programmierung zählen u.a. die Lineare Programmierung, Nicht-Lineare Programmierung und die Integer Programmierung. Für Probleme der Linearen Programmierung, d.h. für die Lösung einer zu minimierenden, linearen Zielfunktion unter Berücksichtigung linearer Nebenbedingungen mit nicht-negativen, reellen Variablen, gibt es Algorithmen, welche diese Probleme lösen können. Integer Programmierung ist eine Erweiterung der Linearen Programmierung, unter der Prämisse, dass Variablentypen ganzzahlige Werte annehmen müssen. Diese Ganzzahligkeit führt dazu, dass diese Problemklasse NP-complete ist, d.h. es wird allgemein angenommen, dass kein allgemeingültig einsetzbarer Algorithmus existiert oder jemals existieren wird, mit dem diese Probleme gelöst werden können. Die Lösung von Problemen der Integer Programmierung ist daher mit sehr hohen Berechnungskosten verbunden. Ähnlich verhält es sich mit Problemen der Nicht-Linearen Programmierung. Hier werden nicht-lineare Zielfunktionen oder Nebenbedingungen deklariert, welche die Lösung dieser Probleme extrem aufwändig gestalten. Alle Problemklas-

sen finden sich auch im Bereich der in dieser Arbeit behandelten Grundrissplanung in der Architektur.

OPL ist eine Constraint Programmiersprache. Sie gleicht in ihrem Aufbau den älteren Constraint Programmiersprachen wie Prolog, CHIP oder CLP. Der Begriff Programmierung bezieht sich hier auf das Paradigma der Programmiersprache (z. B.: Objektorientierte, Prozedurale, Logische Programmiersprache). Die Grundlage von OPL ist eine 2-stufige Architektur, die aus einer Constraint-Komponente und einer Programmier-Komponente besteht. Die Aufgabe der Constraint-Komponente ist die Überprüfung der Erfüllbarkeit der spezifizierten Nebenbedingungen des Optimierungsproblems. In Anlehnung an den „Memory Store“ traditioneller Programmiersprachen, wird die Constraint-Komponente daher auch „Constraint-Store“ genannt. Die Constraint-Komponente ist eingebettet in eine Programmierkomponente, mit welcher grundlegende Operationen, die auf den Constraint-Store Anwendung finden sollen, beschrieben werden können. Wegen dieses Aufbaus besteht die Vorgehensweise zur Lösung kombinatorischer Optimierungsaufgaben mit OPL aus zwei grundlegenden Teilen: Die Festlegung der zu erfüllenden Nebenbedingungen (Constraint-Teil) und die Beschreibung eines Weges zur Lösungssuche (Search-Teil). Der wesentliche Unterschied im Vergleich zu traditionellen Programmiersprachen liegt in der Deklaration der Constraints. Diese müssen als Restriktionen des Lösungsraumes verstanden werden, d.h. Constraints beschreiben, welche Bedingungen eine Lösung erfüllen muss, ohne zu definieren, wie diese Lösung gefunden werden kann. Der Search-Teil dieses Aufbaus ermöglicht die zusätzliche Reduktion des Suchraumes durch die Vorgabe weiterer Restriktionen zur Lösungssuche. Diese Fähigkeit ist von größter Bedeutung zur Lösung harter kombinatorischer Probleme der Integer- oder Nicht-Linearen Programmierung. Aus der Trennung von Constraint-Teil und Search-Teil resultiert auch die Möglichkeit, unterschiedliche Suchmethoden auf einen Constraint-Store anzuwenden, ohne das Programm neu schreiben zu müssen. Daneben ist die Sprache so aufgebaut, dass eine klare Trennung zwischen Modellbeschreibung und Datendeklaration erfolgen kann. Auf diese Weise ist sichergestellt, dass ein Modell mit unterschiedlichen Datensätzen verwendet werden kann. Viele Probleme der kombinatorischen Optimierung können nicht mittels algebraischer Notation beschrieben werden. Zu diesen Problemen gehören insbesondere die Bereiche Terminplanung (Scheduling) und Ressourcen-Verteilung (Resource Allocation). Um derartige Klassen von Problemen lösen zu können, wurden in OPL zahlreiche neue Datentypen (z. B. Activities, ScheduleOrigin, ScheduleHorizon, Reservoirs, u.a.) eingeführt. Experimente mit diesen Datentypen zeigten, dass selbst auf Geometrie basierende Probleme der Grundrissplanung auf diese Weise erfolgreich gelöst werden können. OPL ist der einzige Prototyp, mit welchem sowohl die Umsetzung des Zerstörungsreichen als auch des Zerstörungssarmen Modells realisiert werden konnte.

Die Berechnungszeiten einer numerischen oder graphischen Lösung spielen in der Praxis eine gewichtige Rolle. Für die Bewertung der Funktionsfähigkeit der Prototypen, sind die genannten Berechnungszeiten jedoch von untergeordneter Bedeutung. Die Berechnungsgeschwindigkeit ist von zu vielen unterschiedlichen Faktoren abhängig, als dass mit den durchgeführten Berechnungen Aussagen über verallgemeinerbare Laufzeiteigenschaften getroffen werden könnten. Derartige Laufzeiteigenschaften könnten ausschließlich unter genormten Laborbedingungen evaluiert werden, welche bislang aber nicht existieren. Auf die Praxis hätten derartige Ergebnisse jedoch wiederum wenig Einfluss, da naturgemäß die Ausstattung und Installation der einge-

setzten Rechner stark divergiert. Wesentlich aussagekräftiger ist der Vergleich der Berechnungsergebnisse mit und ohne Verwendung von Such-Methoden, da hier die Laufzeiteigenschaften erhebliche Unterschiede aufweisen. Exemplarisch seien drei Faktoren genannt, die Einfluss auf die Berechnungszeiten haben:

1. Als programminterner Faktor spielt die Verwendung der graphischen Anzeige der Prototypen eine wichtige Rolle. Unter Verwendung der „board“-Methode, die für die graphische Anzeige zwingend notwendig ist, können zur Berechnung nur bestimmte Solver verwendet werden. Diese Solver reduzieren die Berechnungsgeschwindigkeit erheblich. So ist es nicht ungewöhnlich, dass Solver MIP, der die „board-Methode“ unterstützt, im Vergleich zu CPLEX MIP mehr als die zehnfache Berechnungszeit benötigt. Setzte man die Prototypen in industrieller Software ein, würde eine Entkopplung der graphischen Anzeige vom Berechnungsvorgang erfolgen. Mit Hilfe der Schnittstellen zu Java und C++, welche die Entwicklungsumgebung ILOG OPL Studio zur Verfügung stellt, wäre dies problemlos zu realisieren.
2. Updates der OPL-Version 3.7 auf Version 3.7.1 erbrachten teilweise Geschwindigkeitsverbesserungen von Faktoren bis zu 200. Dies kann jedoch nicht für alle Betriebssystemplattformen bescheinigt werden. Die verwendeten Versionen sind zwar unter verschiedenen Betriebssystemen lauffähig, Versuche unter Linux wurden jedoch auf Grund des unsicheren und langsamen Laufzeitverhaltens abgebrochen.
3. In vielen Fällen werden von den Prototypen Berechnungen durchgeführt, bei denen versucht wird, den Lösungsraum vollständig zu erfassen. Dies führt mitunter dazu, dass bereits nach wenigen Sekunden eine perfekte Lösung erzielt wird, der gesamte Berechnungsvorgang jedoch noch nicht abgeschlossen ist.

Es ist ferner anzumerken, dass sowohl in der Praxis als auch in der Forschung, während der Berechnungen andere Aufgaben vom Computer übernommen werden (Netzwerkverbindungen, Virens Scanner, Firewall, TSR-Programme, o.ä). Es ist daher nur bedingt möglich, die erzielten Berechnungszeiten zu bewerten. Als Anhaltspunkt seien hier die arithmetischen Mittel aus drei Dhrystone- und Whetstone-Berechnungen des verwendeten Rechners (Pentium M, 1.7 GHz, 1.5 GB RAM) unter Praxisbedingungen genannt:

- Dhrystone: 7211 MIPS
- Whetstone: 3011 MFLOPS

### 7.3.3 Exemplarische Ergebnisse (Zerstörungsreiches Modell)

Die nachfolgenden Beispiele zeigen die Anwendung des Prototyps auf rechteckige Gebäudegrundrisse und Units des Zerstörungsreichen Modells.

#### 7.3.3.1 Rechteckige Grundrisse und Units

Beispiel 1 zeigt die Anwendung des Prototyps auf das Tong-Problem. Die grundsätzliche Funktionsfähigkeit des mathematischen Modells (vgl. Kap. 6.1.3) und des Prototyps wird an diesem Beispiel (Constraint Satisfaction) geringer Komplexität dargelegt.

Beispiel 2 veranschaulicht die Lösung des Pfefferkorn-Problems, dessen mittlere Komplexität mehr als vier Milliarden potentielle Lösungsmöglichkeiten beinhaltet. Das Problem (Constraint Satisfaction) wird in weniger als einer Sekunde gelöst.

Beispiel 3 verdeutlicht die Anwendung des Prototyps auf ein Maximierungsproblem mittlerer Komplexität unter Verwendung eines weiteren mathematischen Modells (vgl. Kap. 6.1.2). Das Beispiel veranschaulicht die Unterschiede der numerischen und graphischen Berechnungskosten. Ferner wird die Funktionalität der Design Constraints demonstriert, die hier auf Symmetrien und Seitenverhältnisse zwischen verschiedenen Units angewendet werden.

Ein Problem (Constraint Satisfaction) geringer Komplexität aus der Planungspraxis wird in Beispiel 4 demonstriert. Die Ergebnisse der Problemlösung werden der manuellen Entwurfsbearbeitung gegenübergestellt. Es wird das mathematische Modell aus Kap. 6.1.2 verwendet. Ferner werden Search-Procedures eingeführt, mit deren Hilfe die Ausrichtung der Units an einem existenten Stützenraster erfolgen kann. Zur Einschränkung von Flächengrößen werden Wertebereiche für Obere und Untere Schranken definiert.

Beispiel 5 zeigt, dass ein vermeintlich triviales architektonisches Problem ein sehr komplexes Optimierungsproblem (Constraint Satisfaction) darstellen kann. An Hand des Maculet-Problems wird verdeutlicht, dass die kombinatorische Explosion dazu führt, dass ein Problem auf praxisüblichen Rechnern ohne die Verwendung von Search-Procedures nicht lösbar ist. Das Beispiel verdeutlicht ebenfalls die Vereinfachung des OPL-Programmcodes mit Hilfe der entwickelten ConnectionSets.

Aufbauend auf den Erkenntnissen aus Beispiel 5, zeigen die Beispiele 6a – 6c Lösungen des Maculet-Problems unter Verwendung unterschiedlicher Search-Procedures. Es wird dargelegt, wie groß die potentielle architektonische Varianz bei der Erzeugung von Lösungen für dieses Problem sein kann. Die Beispiele verdeutlichen in besonderem Maße, dass maschinelle Entwurfsprozesse den manuellen Prozessen vor allen Dingen in der Erzeugung einer großen Anzahl alternativer Lösungen eines Problems überlegen sind. Sofern nicht explizit angegeben, verwendet der Prototyp bei den obigen Beispielen das mathematische Modell aus Kap. 6.1.3.

## Beispiel 1

### Datei

7331\_1.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.3)

### Zielfunktion

Vier rechteckige Flächen sollen innerhalb einer  $9 \times 9$  m großen Grundfläche angeordnet werden, so dass die Anordnung der Flächen die nachfolgend genannten Nebenbedingungen erfüllt (Constraint Satisfaction). Diese Problemsstellung wurde 1975 von Christopher Tong entwickelt und ist in der Literatur [Medjdoub und Yannou, 2001] als Tong-Problem erwähnt.

### Nebenbedingungen

- Alle Flächen befinden sich innerhalb der quadratischen Grundfläche (FInside).
- Keine der Flächen darf eine andere Fläche überschneiden (PInter).
- Die Seitenlängen der 4 Flächen liegen zwischen 4 und 9 m (Length Constraint).
- Die 4 Flächen füllen die quadratische Grundfläche vollständig aus (Design Constraint).

### Berechnungszeit

Numerische Lösung: 00,90 Sek. (Solver)

Graphische Lösung: 07,80 Sek. (Solver)

Anzahl der Lösungen: 288

### Lösung

Abb. 7.9 zeigt exemplarisch 8 von 288 ermittelten Lösungen für das Tong-Problem.

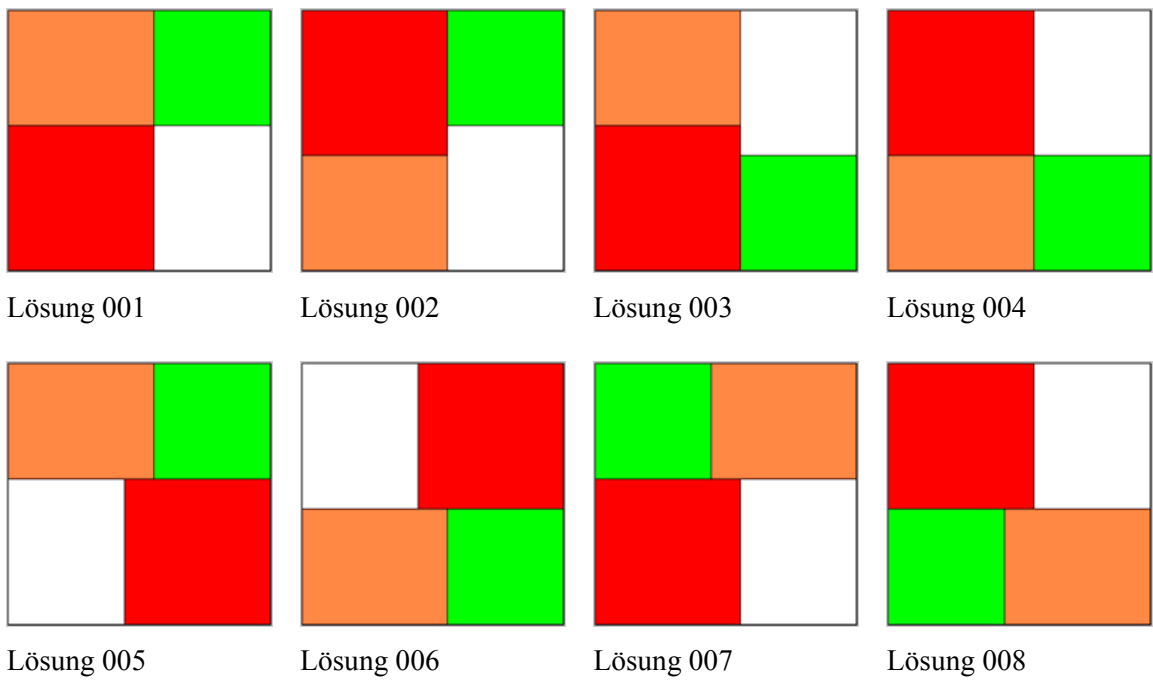


Abb. 7.9 Lösungen des Tong-Problems



## Beispiel 2

### Datei

7331\_2.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.3)

### Zielfunktion

6 Flächen mit konstanten Abmessungen sollen innerhalb einer 8 x 5 m großen Grundfläche angeordnet werden, so dass die Anordnung der Flächen die nachfolgend genannten Nebenbedingungen erfüllt (Constraint Satisfaction). Diese Problemsstellung ist in der Literatur als Pfefferkorn-Problem erwähnt [Pfefferkorn, 1975].

### Nebenbedingungen

- Alle Flächen befinden sich innerhalb der rechteckigen Grundfläche (FInside).
- Keine der Flächen darf eine andere Fläche überschneiden (PInter).
- Die Seitenlängen der 6 Flächen betragen 6x2, 4x2, 2x3, 2x3, 2x3 und 2x1m (Length Constraint).
- Die 6 Flächen füllen die rechteckige Grundfläche vollständig aus (Design Constraint).

### Berechnungszeit

Numerische Lösung: 00,95 Sek. (Solver MIP)

Graphische Lösung: 08,86 Sek. (Solver MIP)

Anzahl der Lösungen: 24

### Lösung

Die Größe des Suchraumes für das Pfefferkorn-Problem beträgt  $(8 \times 5)^6 = 4.096.000.000$  potentielle Lösungsmöglichkeiten. Abb. 7.10 zeigt die 24 möglichen Lösungen, die alle Nebenbedingungen erfüllen. Die Berechnungszeit beträgt lediglich 00,95 Sekunden.

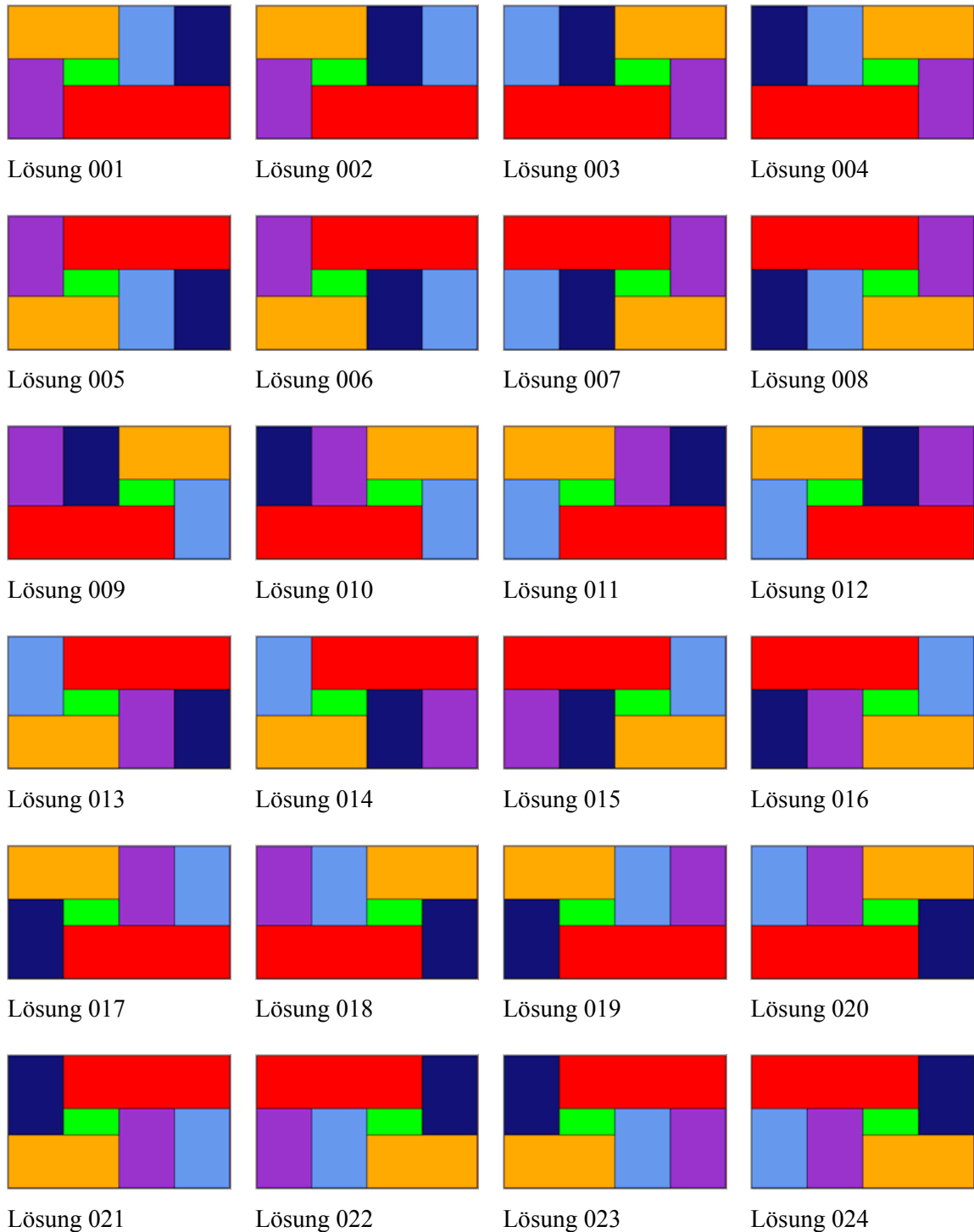


Abb. 7.10 Lösungen des Pfefferkorn-Problems

## Beispiel 3

### Datei

7331\_3.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.2)

### Zielfunktion

Gegeben ist eine Freifläche innerhalb eines Gebäudegrundrisses mit den Abmessungen 14 x 14 m. Finde eine Raumanordnung für 9 Räume, so dass der Umfang der Räume Rot, Blau, Gelb, Orange, Mitternachtsblau und Indisch Rot ( $i - n$ ) ein Maximum erreicht und die nachfolgend genannten Bedingungen erfüllt sind:

### Nebenbedingungen

- Alle Räume befinden sich innerhalb des quadratischen Gebäudegrundrisses (FInside).
- Keiner der Räume darf einen anderen Raum überschneiden (PInter).
- Raum Rot berührt Raum Blau, Gelb und Orange (FConn).
- Raum Mitternachtsblau berührt Raum Gelb (FConn).
- Raum Blau und Rot, sowie Mitternachtsblau und Indisch Rot sind achsensymmetrisch zu Raum Gelb (Symmetry Constraint).
- Die Seitenlängen der Räume Rot und Blau liegen zwischen 2 und 4m (Length Constraint).
- Die Seitenlängen der Räume Gelb, Mitternachtsblau, Indisch Rot, Weiß, Cyan und Schwarz sind größer als 2m (Length Constraint).
- Raum Orange ist quadratisch mit einer Seitenlänge von 3m (Aspect Ratio Constraint).
- Raum Cyan ist doppelt so breit wie Raum Schwarz (Aspect Ratio Constraint).

### Berechnungszeit

Numerische Lösung: 510,91 Sek. (Solver + Hybrid)

Graphische Lösung: 11.680,73 Sek. (Solver + Hybrid + Scheduler)

### Lösung

Die Ergebnisse zeigen die Anordnung der 9 Räume innerhalb des quadratischen Gebäudegrundrisses (Abb. 7.11). Die Maximierung der Raumumfänge führt zu unterschiedlichen Grundrissanordnungen, wobei alle Nebenbedingungen in den jeweiligen Ergebnissen berücksichtigt und eingehalten werden. Die Verschiebung der Symmetrie (vertikale Symmetrieachse / schwarze

Linie) zwischen den Flächen Gelb, Mitternachtsblau und Indisch Rot bei einem Umfang von 122m ist mathematisch korrekt. Sie ist auf die Verwendung von Integer-Werten zurück zu führen, welche für die graphische Anzeige mit Hilfe der board-Methode notwendig sind. Die Breite der gelben Fläche ist ein ungerader Integer-Wert, so dass eine Division durch 2, zur Ermittlung der Symmetrie, einen Integer-Wert zurückliefert, der in jedem Fall ungleich dem rechnerisch richtigen Float-Wert ist. Die Verlagerung der graphischen Anzeige in externe Applikationen und die Verwendung von Float-Variablen würde hier zu graphisch eindeutigen Ergebnissen führen.

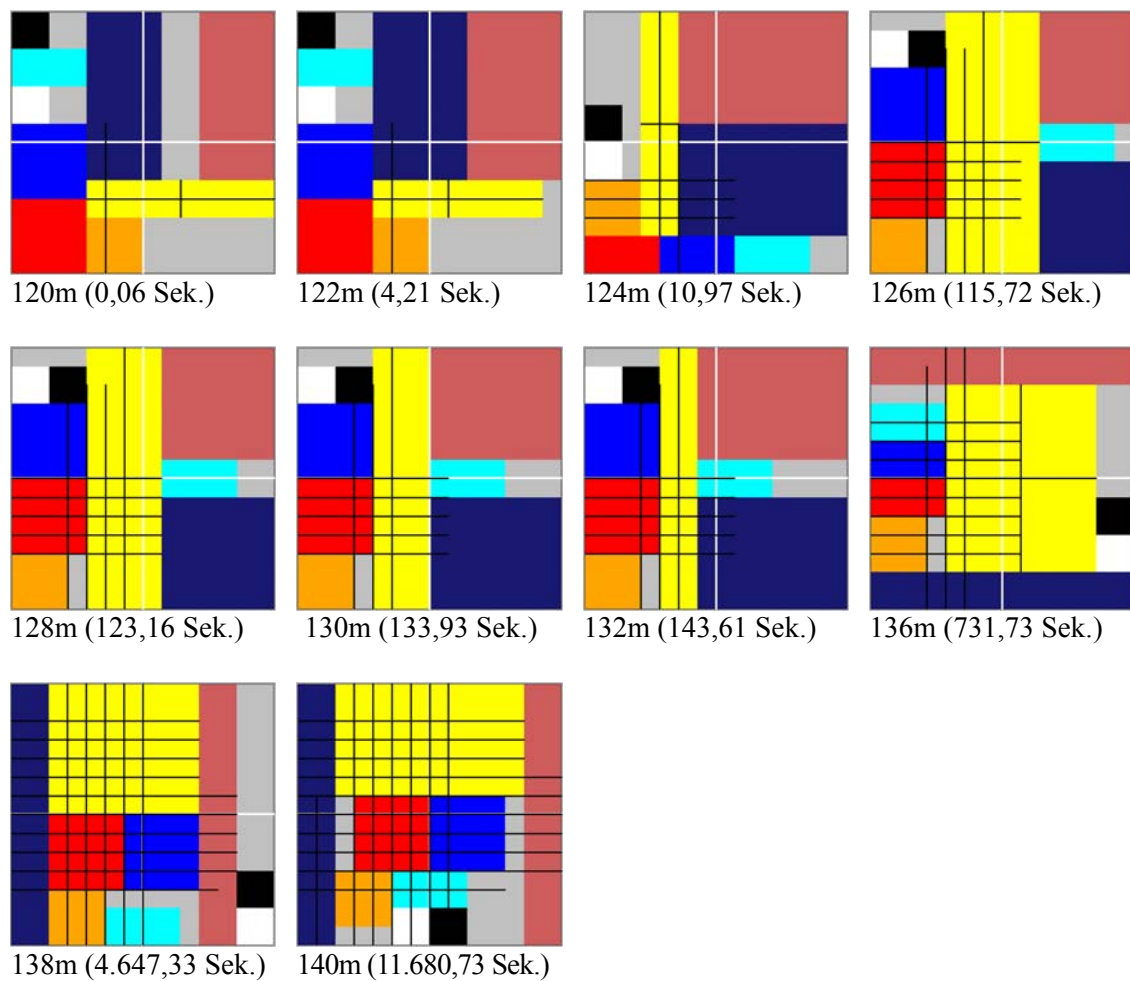


Abb. 7.11 Ergebnisse der Umfangmaximierung

## Beispiel 4

### Datei

7331\_4.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.2)

### Zielfunktion

Gegeben ist ein rechteckiger Gebäudegrundrisses (Eiermann-Bau Apolda, 1. Obergeschoss). Finde eine Raumanordnung für die 7 Räume des Raumprogramms, so dass die Anordnung der Räume die nachfolgend genannten Nebenbedingungen erfüllt (Constraint Satisfaction).

### Nebenbedingungen:

Die Nebenbedingungen werden aus den Flächenerfordernissen des Raumprogramms gebildet. Die Berechnung der Flächengrößen wird unter Verwendung Oberer und Unterer Schranken durchgeführt.

Nummer	Raum	Farbe	Fläche
(Nr. 0)	Grundfläche	dunkelgrau	1032 m <sup>2</sup>
(Nr. 1-4)	Studio / Büro 1-4	grün	4 x 119 m <sup>2</sup>
(Nr. 5)	Kommunikationsbereich	weiß	110 m <sup>2</sup>
(Nr. 6)	Fachbibliothek	sienna1	178 m <sup>2</sup>
(Nr. 7)	Industriearchiv	sienna1	297 m <sup>2</sup>

Folgende Nebenbedingungen werden auf die Flächen des Raumprogramms angewendet:

- Alle Räume befinden sich innerhalb des rechteckigen Gebäudegrundrisses (FInside).
- Keiner der Räume darf einen anderen Raum überschneiden (PInter).
- Alle Räume müssen eine Mindestbreite und Maximallänge aufweisen, die eine dem Raumprogramm entsprechende Nutzung zulassen (Length Constraint).
- Verbindungen zwischen Räumen müssen die Mindestbreite einer Türöffnung aufweisen (Design Variablen).
- Verbindungen zu Außenwänden müssen eine Mindestbreite aufweisen, die eine natürliche Belichtung zulassen (Design Variablen).
- Das Studio / Büro 1 muss eine Verbindung zum Treppenhaus-West besitzen (FConn).

- Die 4 Studio- / Büroräume müssen Zusammenliegen, d.h. 3 Verbindungen besitzen (FConn).
- Das Studio / Büro 1 muss eine Verbindung zum Kommunikationsbereich besitzen (FConn).
- Der Kommunikationsbereich muss eine Verbindung zur Fachbibliothek besitzen (FConn).
- Die Fachbibliothek muss eine Verbindung zum Industriearchiv besitzen (FConn).
- Das Industriearchiv muss eine Zuwegung zum Treppenhaus-Nord besitzen (FConn).
- Die 4 Studio- / Büroräume müssen natürlich belichtet werden können (F2Border).

### Berechnungszeit

Numerische Lösung: -  
 Graphische Lösung: 00,07 Sek. (Solver + Scheduler)

### Lösung

Die dunkelgraue Fläche (Abb. 7.12) zeigt den rechteckigen Gebäudegrundriss des 1. Obergeschosses des Eiermann-Baus in Apolda. In der darauf folgenden Abbildung ist die vollständige Erfüllung aller Nebenbedingungen dargestellt (Abb. 7.13). Zwischen der Fachbibliothek und der Außenwand ist jedoch ein Freiraum erkennbar. Die Schließung dieses Freiraumes erfolgt durch die Verwendung der in OPL implementierten Search-Procedures (Abb. 7.14). Deren Aufgabe ist es, den zu durchsuchenden Lösungsraum eines Optimierungsproblems einzuschränken. Dafür wird ein Operator ( $_{mod}$ ) verwendet, der in diesem Fall die Längen- und Breiten-Variablen der Räume auf Vielfache von  $n$  begrenzt. Mit Hilfe dieses Konstruktes ist es möglich, die Räume des Raumprogramms an bestehenden Stützenrastern auszurichten, um möglichst wenig Eingriffe in den Bestand durchführen zu müssen. Abb. 7.15 zeigt im Vergleich die manuelle Entwurfslösung zweier Architektinnen [Marquardt und Raabe, 2002].



Abb. 7.12 Rechteckige Grundrissform (dunkelgraue Fläche)



Abb. 7.13 Flächenneuordnung ohne Search-Procedure, Eiermann-Bau Apolda, 1.OG

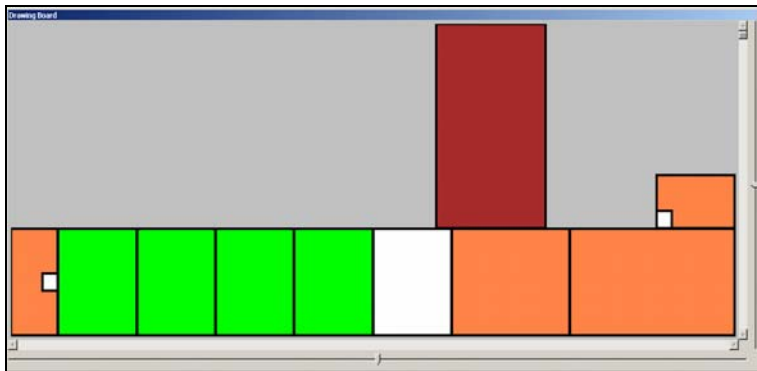


Abb. 7.14 Flächenneuordnung mit Search-Procedure, Eiermann-Bau Apolda, 1. OG

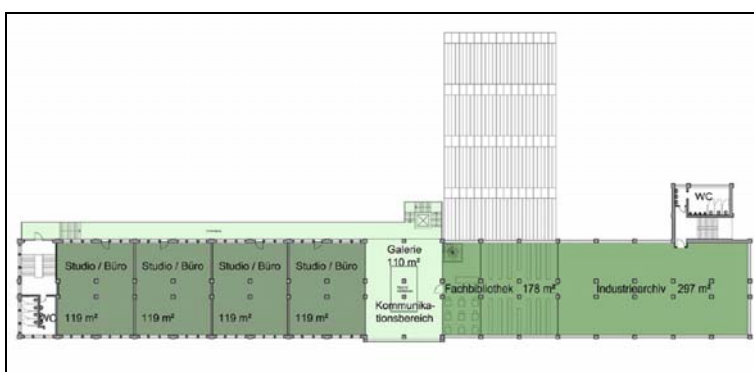


Abb. 7.15 Manuelle Entwurfslösung, Eiermann-Bau Apolda, 1. OG [Marquardt und Raabe, 2002]



Abb. 7.16 Eiermann-Bau Apolda [InfAR, 2005]



## Beispiel 5

### Datei

7331\_5.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.3)

### Zielfunktion

Gegeben ist ein  $120\text{m}^2$  großer rechteckiger Gebäudegrundriss mit den Seitenlängen 12 und 10 m. Finde innerhalb dieses Grundrisses alle möglichen Raumanordnungen für die 10 Räume des Raumprogramms, so dass die Anordnung der Räume die nachfolgend genannten Nebenbedingungen erfüllt (Constraint Satisfaction). Diese Problemsstellung wurde 1991 entwickelt und ist als Maculett-Problem in der Literatur erwähnt. [Maculett, 1991; nach: Medjdoub und Yannou, 2001].

### Nebenbedingungen

Die Nebenbedingungen werden aus den Flächenerfordernissen des Raumprogramms gebildet. Die Berechnung der Flächengrößen wird unter Verwendung Oberer und Unterer Schranken durchgeführt.

Nummer	Raum	Farbe	F (min – max)	L (min)	B (min)
(Nr. 0)	Grundfläche	grau	120 m <sup>2</sup>	12m	10m
(Nr. 1)	Wohnraum	rot	33 – 42 m <sup>2</sup>	4m	4m
(Nr. 2)	Küche	blau	09 – 15 m <sup>2</sup>	3m	3m
(Nr. 3)	Dusche	orange	06 – 09 m <sup>2</sup>	2m	2m
(Nr. 4)	Verkehrsfläche1	gelb	01 – 12 m <sup>2</sup>	1m	1m
(Nr. 5)	Verkehrsfläche2	gelb	01 – 12 m <sup>2</sup>	1m	1m
(Nr. 6)	Raum 1	indisch rot	11 – 15 m <sup>2</sup>	3m	3m
(Nr. 7)	Raum 2	sienna1	11 – 15 m <sup>2</sup>	3m	3m
(Nr. 8)	Raum 3	weiß	11 – 15 m <sup>2</sup>	3m	3m
(Nr. 9)	Toilette	cyan	01 – 02 m <sup>2</sup>	1m	1m
(Nr. 10)	Raum 4	olivgrün	15 – 20 m <sup>2</sup>	1m	1m

(Anm.: F = Fläche, L = Länge, B = Breite, min = Minimum, max = Maximum)

Folgende Nebenbedingungen werden auf die Flächen des Raumprogramms angewendet:

- Alle Räume befinden sich innerhalb des rechteckigen Gebäudegrundrisses (FInside).
- Keiner der Räume darf einen anderen Raum überschneiden (PInter).
- Die Gesamtfläche der Räume entspricht der Grundfläche des Gebäudegrundrisses.
- Alle Räume müssen eine Mindestbreite und Mindestlänge aufweisen, die eine dem Raumprogramm entsprechende Nutzung zulassen (Length Constraint).
- Verbindungen zwischen Räumen müssen eine Mindestbreite von 1m aufweisen (Design Variablen).
- Verbindungen zu Außenwänden müssen eine Mindestbreite von 1m aufweisen (Design Variablen).
- Der Wohnraum muss auf der Süd- oder Westseite des Grundrisses angeordnet werden.
- Die Küche muss auf der Süd- oder Nordseite des Grundrisses angeordnet werden.
- Raum 1 muss auf der Süd- oder Nordseite des Grundrisses angeordnet werden.
- Raum 2 muss auf der Süd- oder Nordseite des Grundrisses angeordnet werden.
- Raum 3 muss auf der Süd- oder Nordseite des Grundrisses angeordnet werden.
- Raum 4 muss auf der Südseite des Grundrisses angeordnet werden.
- Alle Räume, mit Ausnahme der Küche, müssen eine der beiden Verkehrsflächen berühren. Die Berührungsbreite muss mindestens 1m betragen.
- Der Wohnraum und die Küche müssen sich berühren.
- Die Küche und die Dusche müssen sich berühren.
- Die Toilette muss die Küche oder die Dusche berühren.
- Die zwei Verkehrsflächen müssen sich berühren.
- Der Eingang in das Gebäude verläuft durch den Wohnraum.

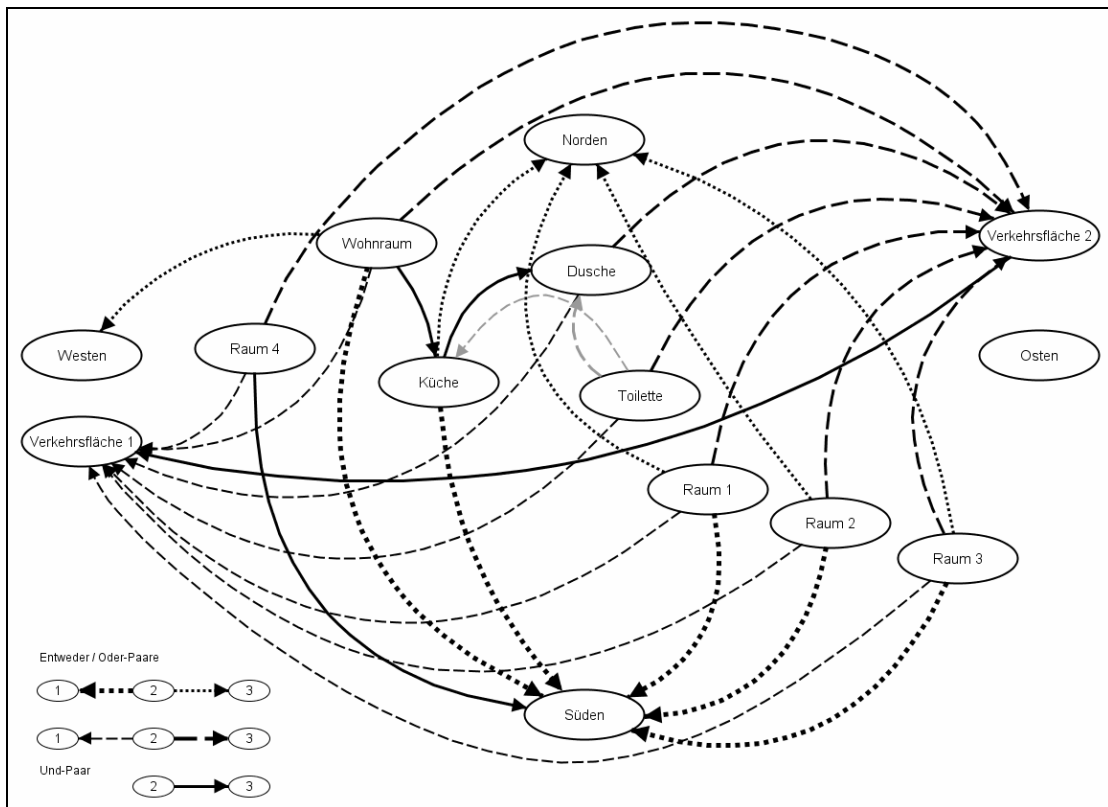


Abb. 7.17 Bubble-Diagramm des Raumprogramms für das Maculet-Problem

### Berechnungszeit

Numerische Lösung: -

Graphische Lösung: 74.069,81 Sek. bei Lösung Nr. 252 (Solver)

Anzahl der Lösungen: Programmabbruch (BlueScreen) bei Lösung Nr. 326

### Lösung

Das Maculet-Problem ist mit extrem hohen Berechnungskosten verbunden. Die berechneten Lösungen verdeutlichen die mathematische Komplexität der architektonischen Problemstellung. Nach 326 Lösungen bricht die Berechnung des Problems mit einem BlueScreen ab. Die letzte dokumentierte Angabe zum Berechnungsverlauf wurde vom Solver bei Lösung Nr. 252 erzeugt. Bis zu diesem Zeitpunkt (74.069 Sekunden) wurden ca. 38 Millionen Choice Points (Anzahl der Berechnungsschritte zur Erzeugung einer Lösung) berechnet. Die Anzahl der erzeugten Lösungen verdeutlicht jedoch auch die Komplexität des auf den ersten Blick trivialen, architektonischen Problems.

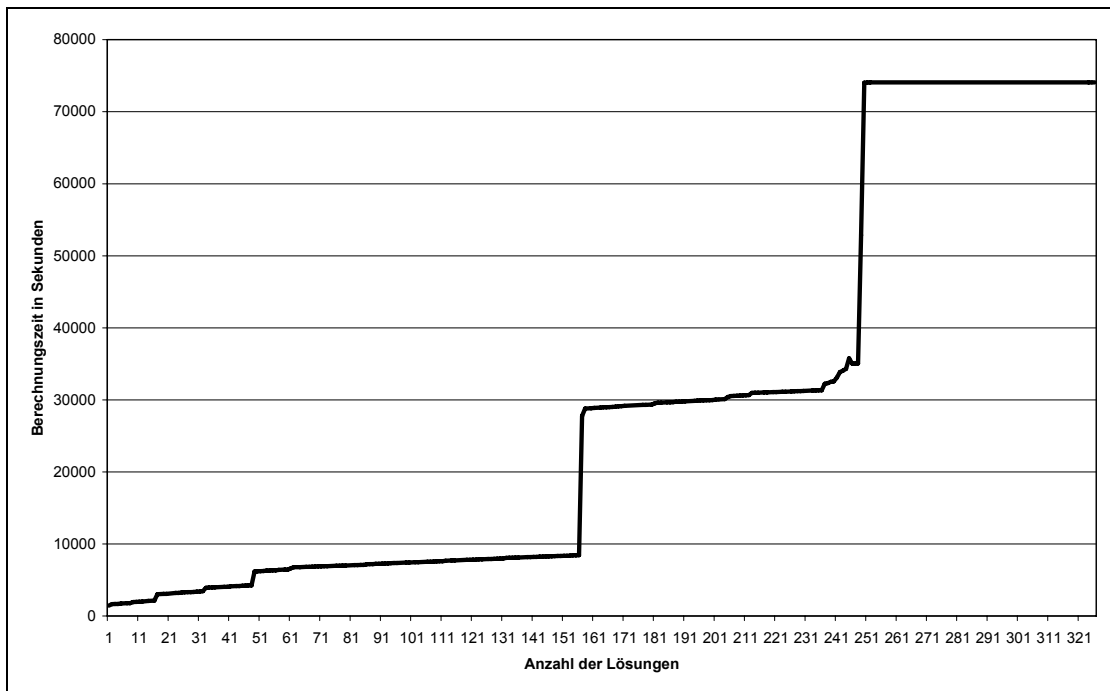


Abb. 7.18 Berechnungszeit des Maculet-Problems

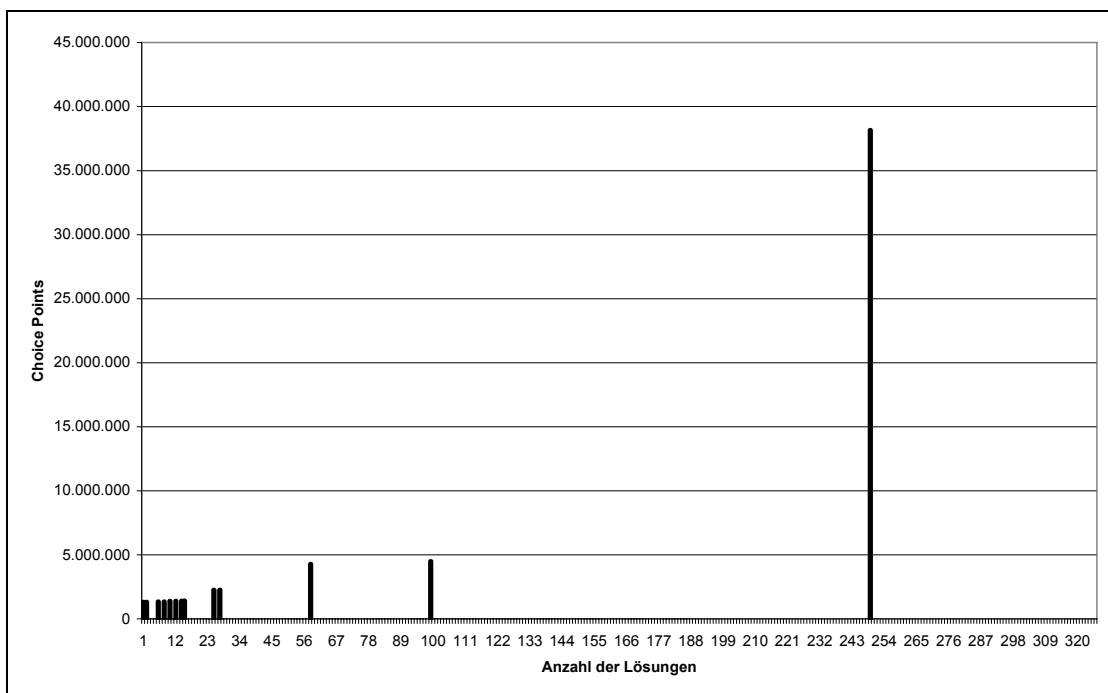


Abb. 7.19 Choice Points des Maculet-Problems

Da die Anzahl der erzeugbaren Lösungen für das Maculet-Problem sehr viel größer sein wird, ist es sinnvoll, den Suchraum weiter einzuschränken. Diese Einschränkung kann architektonisch, wie in Kapitel 3.2.1 (Zeitlicher Ablauf) erörtert, mit Hilfe von fakultativen Entwurfszielen definiert und mathematisch unter Verwendung von Search Procedures effizient gestaltet werden (vgl. Beispiel 6a-c). Abb. 7.20 zeigt 16 exemplarische graphische Lösungen. Die beiliegende CD beinhaltet eine detaillierte Darstellung aller 326 erzeugten Lösungen, sowie den Programmcode der verwendeten Modelle (7331\_6a.mod, 7331\_6b.mod und 7331\_6c.mod).

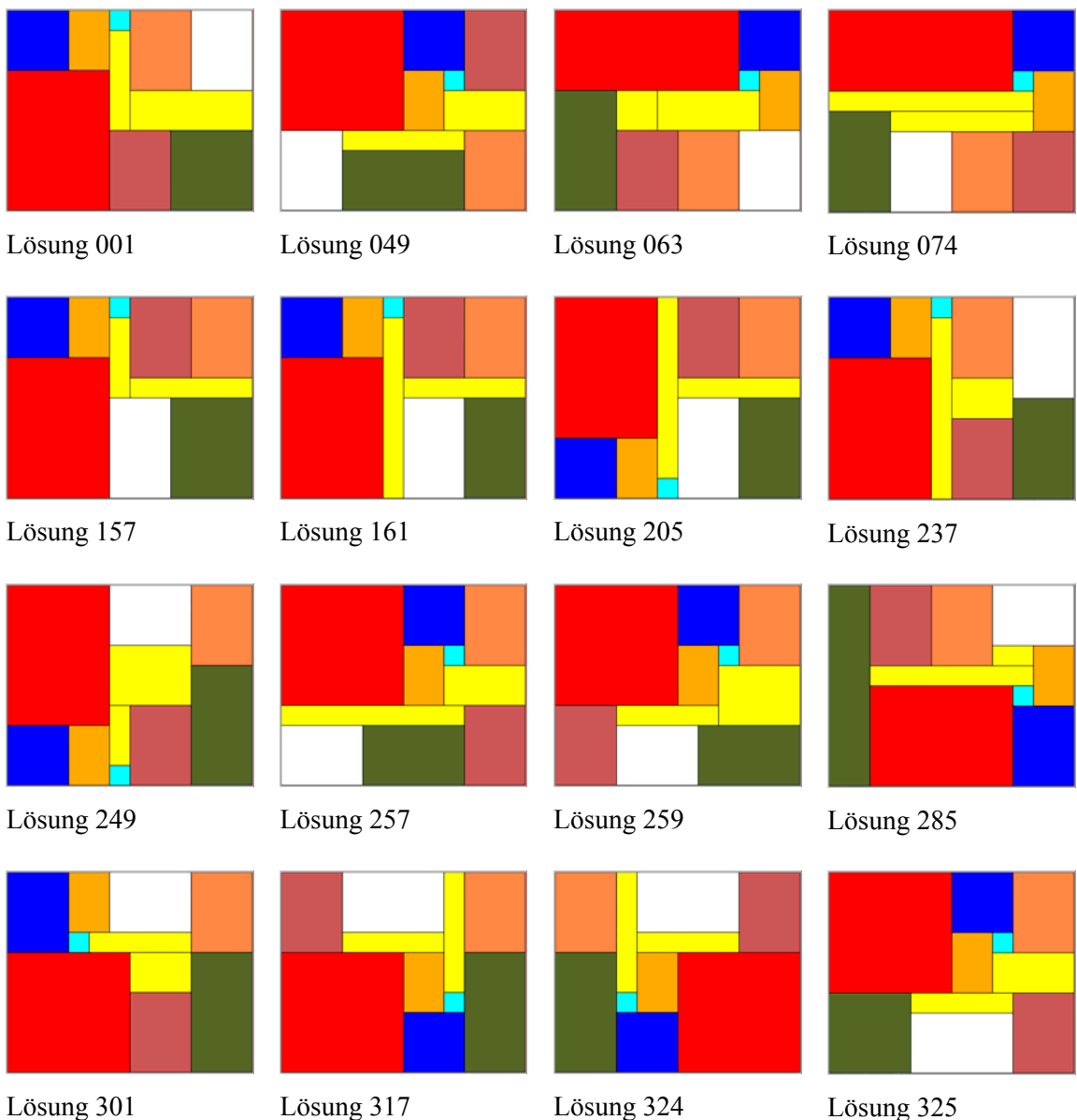


Abb. 7.20 Lösungen des Maculet-Problems

Zur Lösung des Maculet-Problems wurden so genannte ConnectionSets entwickelt, deren Verwendung die Komplexität der programmiertechnischen Umsetzung erheblich reduziert. Eine detaillierte Beschreibung der Einsatzbereiche und Leistungsfähigkeit dieser Sets findet sich in Kap. 7.3.3.2 (Beispiel 2).

## Beispiel 6a

### Datei

7331\_6a.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.3)

### Zielfunktion

Es gilt die in Beispiel 5 (Kap. 7.3.3.1, Datei: 7331\_5.mod) aufgeführte Zielfunktion.

### Nebenbedingungen

Es gelten die in Beispiel 5 (Kap. 7.3.3.1, Datei: 7331\_5.mod) aufgeführten Nebenbedingungen. Zur Einschränkung des Suchraumes, resp. der Ergebnisse, wird folgende Nebenbedingung eingeführt:

- Die Gesamtgröße der Verkehrsfläche wird auf 13 m<sup>2</sup> beschränkt.

Diesem Wert liegt eine Optimierungsberechnung zu Grunde, deren Zielfunktion die Minimierung der Verkehrsfläche war. Um den Lösungsraum nicht auf die kleinstmögliche Größe der Verkehrsfläche einzuschränken, wurde ein Mittelwert verwendet.

### Search Procedure

Es wird zusätzlich eine Search Procedure eingeführt, welche auf die zwei Verkehrsflächen des Raumprogramms angewendet wird:

- Die zwei Verkehrsflächen sind in ihrer Breite in y-Richtung auf 1m beschränkt.

Die Deklaration dieser Einschränkung als Search Procedure, und nicht als Design Constraint, führt dazu, dass der Solver zunächst versucht nur Lösungen zu ermitteln, welche die Wertedeclaration, d.h. die benutzerdefinierte Search Procedure, berücksichtigen. Existiert keine Lösung mit den entsprechenden Wertepaaren, verwendet der Solver eine programminterne Default-Strategie als Search Procedure.

### Berechnungszeit

Numerische Lösung:	-
Graphische Lösung:	43.201,88 Sek. (Solver)
Anzahl der Lösungen:	912

### Lösung

Die Einschränkung des Suchraumes durch die Verwendung der Search Procedure führt zu einer effektiven Verringerung der Berechnungskosten. Innerhalb von 43.201 Sekunden werden 912 mathematisch differente Lösungen ermittelt. Die Anzahl der relevanten Lösungen muss um die Hälfte vermindert werden, da durch das mathematisch korrekte Vertauschen der Verkehrsflächen 1 und 2 architektonisch redundante Lösungen entstehen. Bei Lösung 912 endet die Berechnung nach 13 Millionen Choice Points.

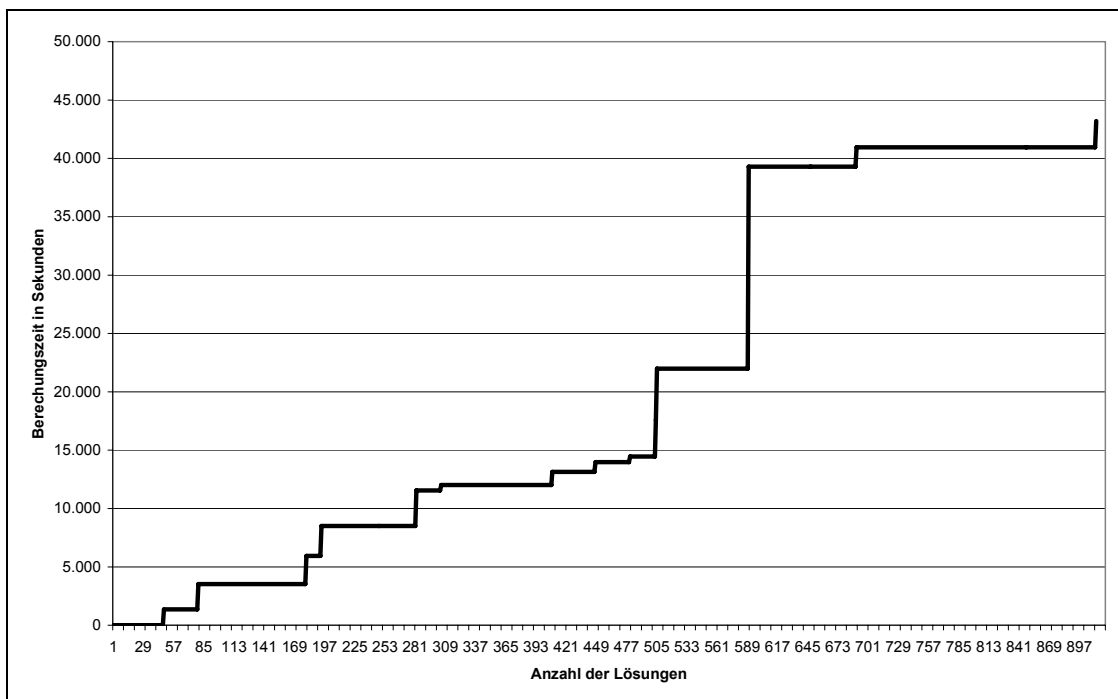


Abb. 7.21 Berechnungszeit des Maculet-Problems mit Search Procedure



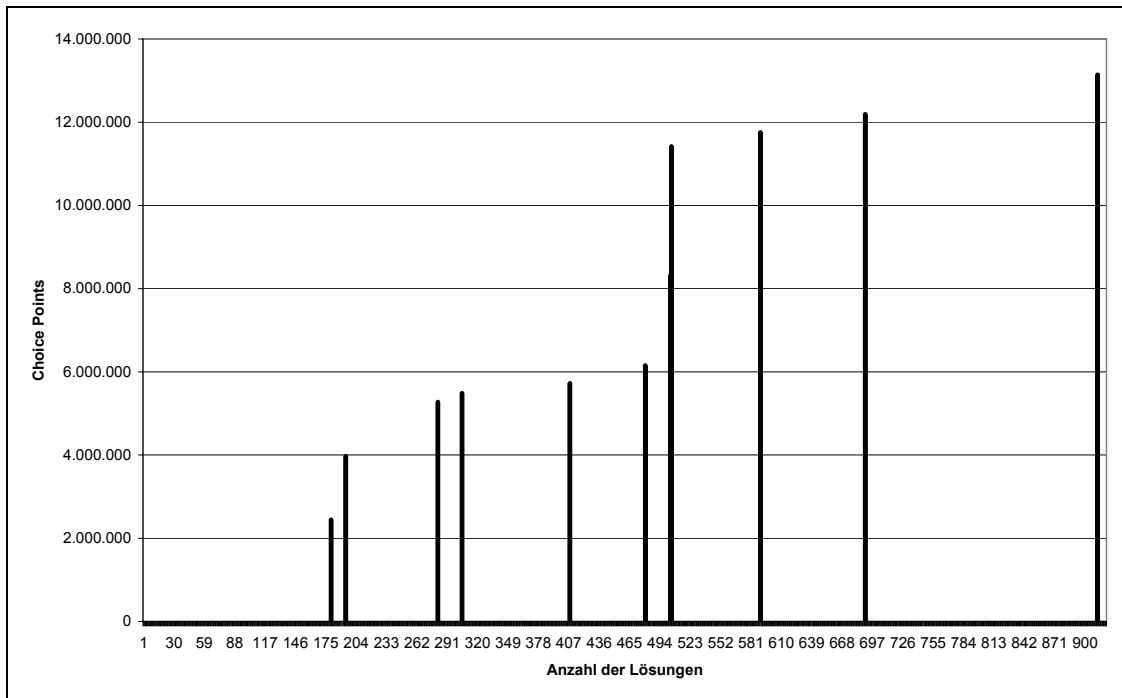
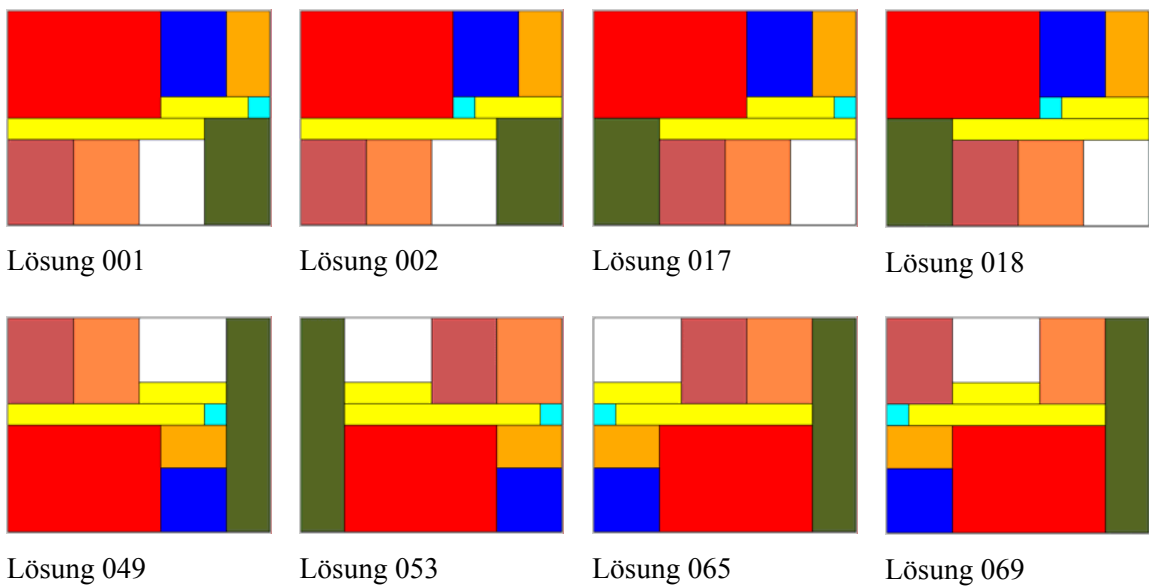
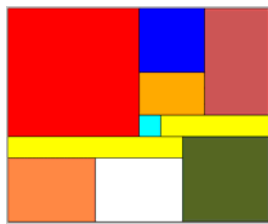


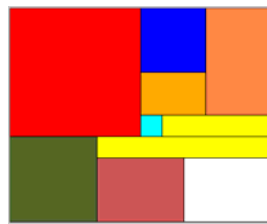
Abb. 7.22 Choice Points des Maculet-Problems mit Search Procedure

Die nachfolgende Abbildung zeigt 16 exemplarische graphische Lösungen (Abb. 7.23). Die beiliegende CD beinhaltet eine detaillierte Darstellung aller 912 erzeugten Lösungen.

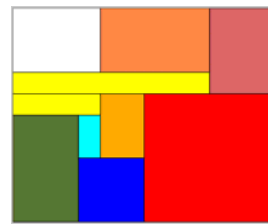




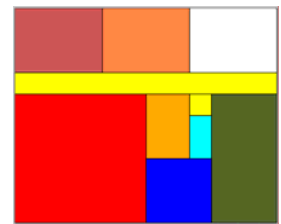
Lösung 121



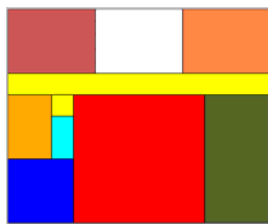
Lösung 169



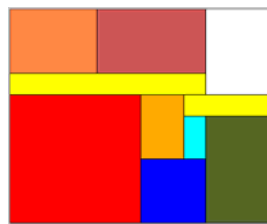
Lösung 253



Lösung 313



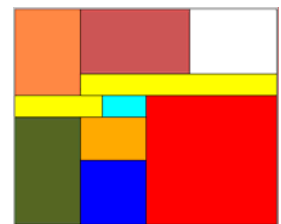
Lösung 345



Lösung 470



Lösung 819



Lösung 843

Abb. 7.23 Exemplarische Lösungen des Maculett-Problems mit Search Procedure (y-Richtung)

## Beispiel 6b

### Datei

7331\_6b.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.3)

### Zielfunktion

Es gilt die in Beispiel 5 (Kap. 7.3.3.1, Datei: 7331\_5.mod) aufgeführte Zielfunktion.

### Nebenbedingungen

Es gelten die in Beispiel 5 (Kap. 7.3.3.1, Datei: 7331\_5.mod) aufgeführten Nebenbedingungen. Zur Einschränkung der Ergebnisse wird folgende Nebenbedingung eingeführt:

- Die Gesamtgröße der Verkehrsfläche wird auf 13 m<sup>2</sup> beschränkt.

Diesem Wert liegt eine Optimierungsberechnung zu Grunde, deren Zielfunktion die Minimierung der Verkehrsfläche war. Um den Lösungsraum nicht auf die kleinstmögliche Größe der Verkehrsfläche einzuschränken, wurde ein Mittelwert verwendet.

### Search Procedure

Es werden zwei zusätzliche Search Procedures eingeführt, welche auf die zwei Verkehrsflächen des Raumprogramms angewendet werden:

- Die erste Verkehrsfläche ist in ihrer Breite in x-Richtung auf 1m beschränkt.
- Die zweite Verkehrsfläche ist in ihrer Breite in y-Richtung auf 1m beschränkt.

Die Deklaration dieser Einschränkung als Search Procedure, und nicht als Design Constraint, führt dazu, dass der Solver zunächst versucht nur Lösungen zu ermitteln, welche die Wertedeklaration, d.h. die benutzerdefinierte Search Procedure, berücksichtigen. Existiert keine Lösung mit den entsprechenden Wertepaaren, verwendet der Solver eine programminterne Default-Strategie als Search Procedure.

### Berechnungszeit

Numerische Lösung:	-
Graphische Lösung:	685,47 Sek. (Solver)
Anzahl der Lösungen:	312

### Lösung

Die Einschränkung des Suchraumes durch die Verwendung der Search Procedure führt zu einer höchst effektiven Verringerung der Berechnungskosten. Innerhalb von lediglich 685 Sekunden werden 312 mathematisch und architektonisch differente Lösungen ermittelt. Bei Lösung 312 endet die Berechnung nach 12 Millionen Choice Points. Die nachfolgende Abbildung zeigt 16 exemplarische graphische Lösungen (Abb. 7.24). Die beiliegende CD beinhaltet eine detaillierte Darstellung aller 312 erzeugten Lösungen.

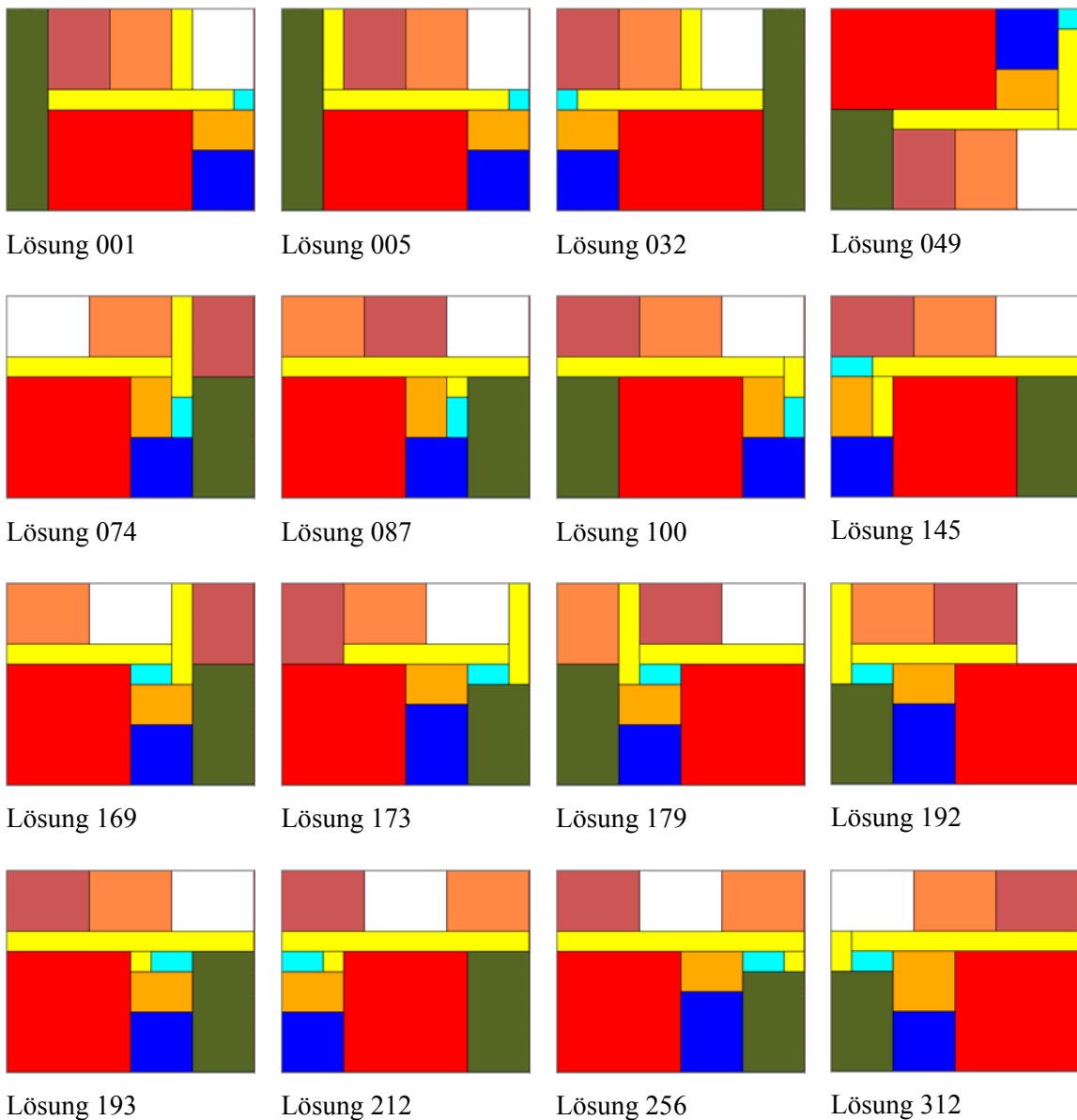


Abb. 7.24 Exemplarische Lösungen des Maculet-Problems mit Search Procedure (x, y-Richtung)

## Beispiel 6c

### Datei

7331\_6c.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.3)

### Zielfunktion

Es gilt die in Beispiel 5 (Kap. 7.3.3.1, Datei: 7331\_5.mod) aufgeführte Zielfunktion.

### Nebenbedingungen

Es gelten die in Beispiel 5 (Kap. 7.3.3.1, Datei: 7331\_5.mod) aufgeführten Nebenbedingungen. Es wird keine zusätzliche Nebenbedingungen eingeführt.

### Search Procedure

Es werden zwei zusätzliche Search Procedures eingeführt, welche auf die zwei Verkehrsflächen des Raumprogramms angewendet werden:

- Die erste Verkehrsfläche ist in ihrer Breite in x-Richtung auf 1m beschränkt.
- Die zweite Verkehrsfläche ist in ihrer Breite in y-Richtung auf 1m beschränkt.

Die Deklaration dieser Einschränkung als Search Procedure, und nicht als Design Constraint, führt dazu, dass der Solver zunächst versucht nur Lösungen zu ermitteln, welche die Wertedeklaration, d.h. die benutzerdefinierte Search Procedure, berücksichtigen. Existiert keine Lösung mit den entsprechenden Wertepaaren, verwendet der Solver eine programminterne Default-Strategie als Search Procedure.

### Berechnungszeit

Numerische Lösung:	-
Graphische Lösung:	14.369,68 Sek. (Solver)
Anzahl der Lösungen:	750

### Lösung

Der Verzicht auf die Verwendung einer Nebenbedingung, welche die Gesamtgröße der Verkehrsflächen restringiert, führt erwartungsgemäß zu einer größeren Anzahl von Lösungen. Da bei diesem Beispiel dennoch Search Procedures verwendet werden (vgl. dazu im Ggs. Beispiel

5, Datei: 7331\_5.mod), kann der Suchraum effektiv und sinnvoll eingeschränkt werden. Innerhalb von 14.369 Sekunden werden 750 mathematisch und architektonisch differente Lösungen ermittelt. Bei Lösung 750 endet die Berechnung nach 342 Millionen Choice Points. Die nachfolgende Abbildung zeigt 16 exemplarische graphische Lösungen (Abb. 7.25). Die beiliegende CD beinhaltet eine detaillierte Darstellung aller 750 erzeugten Lösungen.

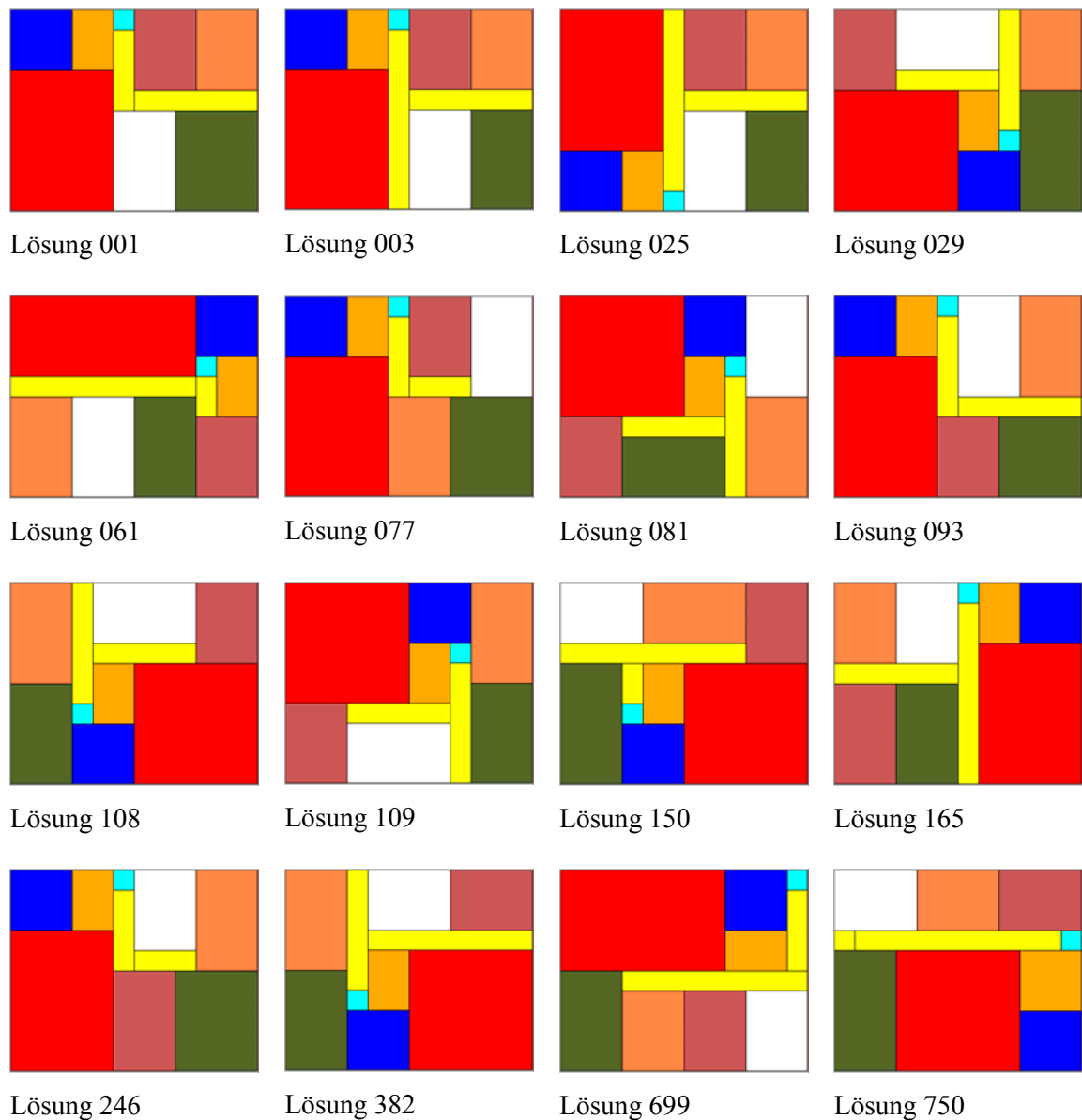


Abb. 7.25 Exemplarische Lösungen des Maculec-Problems ohne Größenbeschränkung der VF

### 7.3.3.2 Rechtwinklige Grundrisse und Units

Rechtwinklige Gebäudegrundrisse werden durch die Verwendung von Void-Units (vgl. Kap. 5.2.1) gebildet. Void-Units werden in den Grafiken durch Auskreuzungen (vgl. Abb. 7.27 - Abb. 7.28) innerhalb der Flächen dargestellt. Rechtwinklige Units werden durch die Verwendung von ConnectionSets (vgl. insb. Kap. 7.3.3.2, Beispiel 2a) erzeugt.

Unter Verwendung Oberer und Unterer Schranken verdeutlicht Beispiel 1 die Anwendung des Prototyps auf ein konkretes Problem (Constraint Satisfaction) aus der Planungspraxis. Die Ergebnisse der Problemlösung werden der manuellen Entwurfsbearbeitung gegenübergestellt. Durch Constraint-Kombinationen werden Void-Units gebildet, mit deren Hilfe rechtwinklige Grundrissformen und Units berechnet werden können.

Die Beispiele 2a-2c zeigen in aufsteigender Komplexität die Anwendung interner und externer ConnectionSets. Beispiel 2a verdeutlicht zudem die Funktionsweise der Flächenapproximation zur Vermeidung quadratischer Zielfunktionen.

Die effektive Anwendung von Search Procedures wird in Beispiel 2b an Hand zweier Varianten mittlerer Komplexität verdeutlicht. Die Beispiele 2a und 2b lösen Probleme zur Maximierung von drei Nutzflächen entlang einer rechtwinkligen Verkehrsfläche.

Beispiel 2c löst ein Constraint Satisfaction Problem unter Berücksichtigung der in 2a und 2b erzielten Ergebnisse. Dabei werden systematisch mehrere Search Procedures eingesetzt, die den Suchraum effektiv eingrenzen und die geometrische Ausprägung der Units an existente Stützraster anpassen. Für das Problem hoher Komplexität werden über 3 Milliarden Berechnungsschritte durchgeführt und über 2800 perfekte Lösungen erzeugt, die sowohl alle Nebenbedingungen als auch die Zielfunktion erfüllen. Die schrittweise Verwendung von Search Procedures verdeutlicht eine potentielle Vorgehensweise des Architekten zur Erzielung perfekter Lösungen für architektonische Optimierungsprobleme des Zerstörungsreichen Modells.

Der Prototyp verwendet in Beispiel 1 das mathematische Modell aus Kap. 6.1.2. Das mathematische Modell aus Kap. 6.1.3 wird in Beispiel 2a-2c verwendet.

## Beispiel 1

### Datei

7332\_1.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.2)

### Zielfunktion

Gegeben ist ein T-förmiger Gebäudegrundrisses (Eiermann-Bau Apolda, Erdgeschoss). Finde eine Raumanordnung für die 13 Räume des Raumprogramms, so dass die Anordnung der Räume die nachfolgend genannten Nebenbedingungen erfüllt (Constraint Satisfaction).

### Nebenbedingungen

Die Nebenbedingungen werden aus den Flächenerfordernissen des Raumprogramms gebildet. Die Berechnung der Flächengrößen wird unter Verwendung Oberer und Unterer Schranken durchgeführt.

Nummer	Raum	Farbe	Fläche
(Nr. 0)	Grundfläche	dunkelgrau	1308 m <sup>2</sup>
(Nr. 1)	Seminarraum 1-3	grün	3 x 47 m <sup>2</sup>
(Nr. 2)	Seminarraum 4	grün	60 m <sup>2</sup>
(Nr. 3)	Seminarraum 5	grün	66 m <sup>2</sup>
(Nr. 4)	Verkehrsfläche	weiß	NN.
(Nr. 5)	Eingang	weiß	129 m <sup>2</sup>
(Nr. 6)	Laden 1-2	see grün	2 x 47 m <sup>2</sup>
(Nr. 7)	Laden 3	see grün	119 m <sup>2</sup>
(Nr. 8)	Kino	braun	217 m <sup>2</sup>
(Nr. 9)	Kino-Vorbereich	braun	119 m <sup>2</sup>
(Nr. 10)	Sportstudio	sienna1	234 m <sup>2</sup>
(Nr. 11)	existentes Treppenhaus-West	weiß	4 m <sup>2</sup>
(Nr. 12)	existentes Treppenhaus-Nord	weiß	4 m <sup>2</sup>



Folgende Nebenbedingungen werden auf die Flächen des Raumprogramms angewendet:

- Alle Räume befinden sich innerhalb des T-förmigen Gebäudegrundrisses (FInside).
- Keiner der Räume darf einen anderen Raum überschneiden (PInter).
- Keiner der Räume darf eine Void-Unit überschneiden (PInter).
- Alle Räume müssen eine Mindestbreite und Maximallänge aufweisen, die eine dem Raumprogramm entsprechende Nutzung zulassen (Length Constraint).
- Verbindungen zwischen Räumen müssen die Mindestbreite einer Türöffnung aufweisen (Design Variablen).
- Verbindungen zu Außenwänden müssen eine Mindestbreite aufweisen, die eine natürliche Belichtung zulassen (Design Variablen).
- Die Seminarräume 1-5 müssen eine Verbindung zur Verkehrsfläche besitzen (FConn).
- Die Verkehrsfläche muss eine Verbindung zum Treppenhaus-West besitzen (FConn).
- Die Läden 1-3 und der Kino-Vorbereich müssen eine Verbindung zum Eingang besitzen (FConn).
- Der Kino-Vorbereich muss eine Verbindung zum Kino besitzen (FConn).
- Das Sportstudio muss eine Zuwegung zum Treppenhaus-Nord besitzen (FConn).
- Die Seminarräume 1-5 und die Läden 1-3 müssen natürlich belichtet werden können (F2Border).

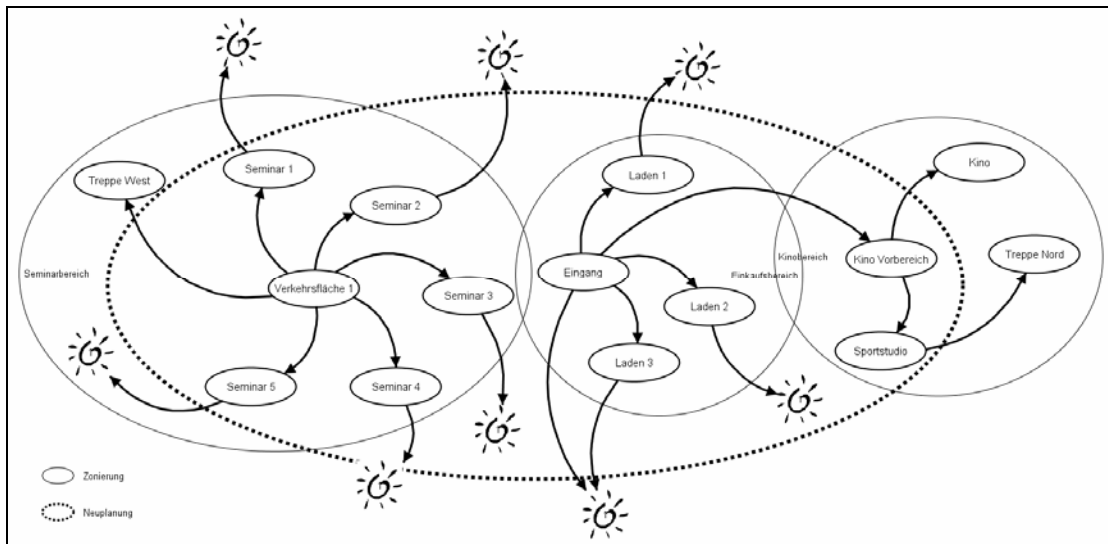


Abb. 7.26 Bubble-Diagramm des Raumprogramms, Eiermann-Bau Apolda, EG

**Berechnungszeit**

Numerische Lösung: -

Graphische Lösung: 01,05 Sek. (Solver + Scheduler)

**Lösung**

Die Ergebnisse zeigen die Grundrissanordnungen unter Verwendung unterschiedlicher Oberer und Unterer Schranken der Design Constraints (Abb. 7.29 - Abb. 7.32). Das erste Ergebnis (Abb. 7.29) zeigt eine punktförmige Berührung zwischen dem Eingang und dem Kino-Vorbereich ohne Berührungsstrecke. Die Größe der Überlappung kann variieren. In der Regel wird das Mindestmaß einer Türöffnung als Überlappungsbereich definiert. Dieses Mindestmaß ist abhängig von der Nutzung der jeweiligen Räume. An Stelle der Überlappung wurde in diesem Fall jedoch ein F2Border Constraint für den Kino-Vorbereich verwendet, welches die Erschließung sicherstellt. Die Abbildungen (Abb. 7.30 - Abb. 7.32) zeigen Varianten des Berechnungsergebnisses ohne Veränderung der ursprünglichen Nebenbedingungen. Die Verwendung von Constraint-Kombinationen zur Erzeugung der Void-Units führt dazu, dass die Seminarräume an einer Außenwand platziert werden können, ohne dass diese als eine solche deklariert worden ist. Eine auf die Void-Units angewandte Disjunktion (ausschließliches Oder) zwischen einem F2Border- und zwei FConn-Constraints sorgt dafür, dass sowohl die ursprünglich definierte Begrenzungsfläche des Gebäudegrundrisses als auch die innen liegenden Seiten der Void-Units als Außenkanten des Gebäudes erkannt werden. Die letzte Abbildung (Abb. 7.33) zeigt im Vergleich die manuelle Entwurfslösung zweier Architektinnen [Marquardt und Raabe, 2002].

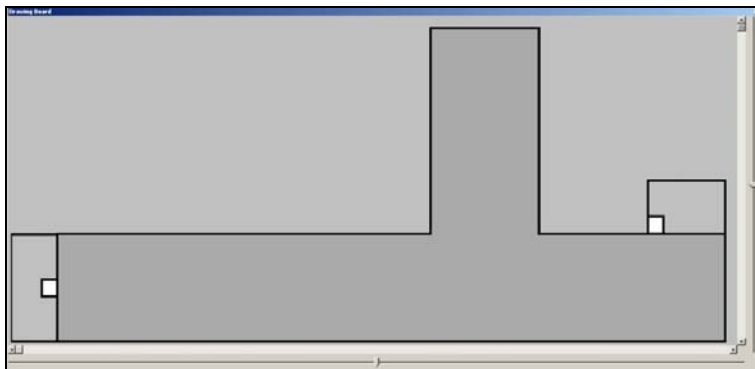


Abb. 7.27 Rechtwinkliger Grundriss (dunkelgrau), Eiermann-Bau Apolda, EG

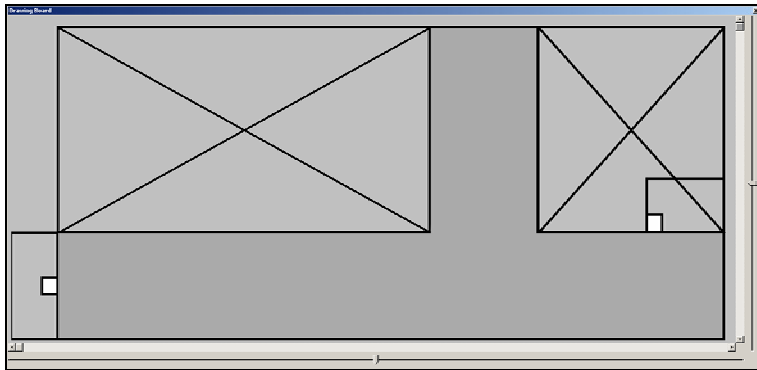


Abb. 7.28 Modellierung einer imaginären rechteckigen Grundrissform durch Void-Units

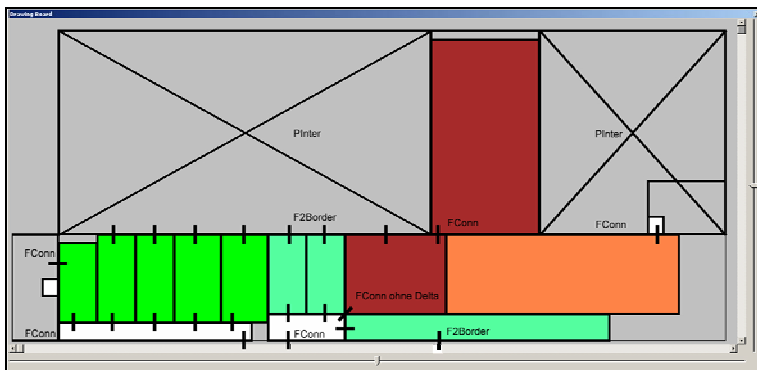


Abb. 7.29 Ergebnis 1 der Flächenneuordnung, Eiermann-Bau Apolda, EG

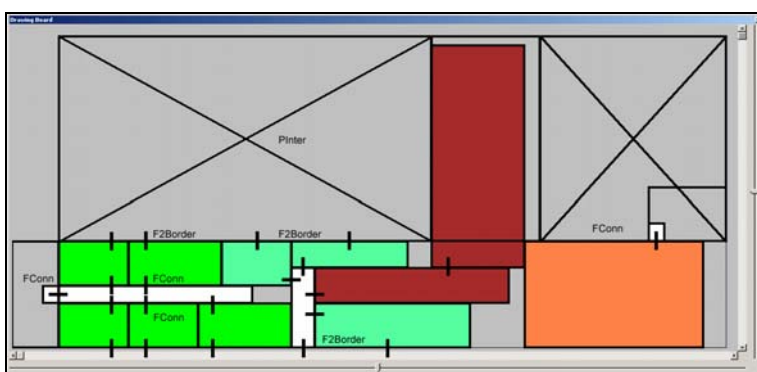


Abb. 7.30 Ergebnis 2 der Flächenneuordnung, Eiermann-Bau Apolda, EG

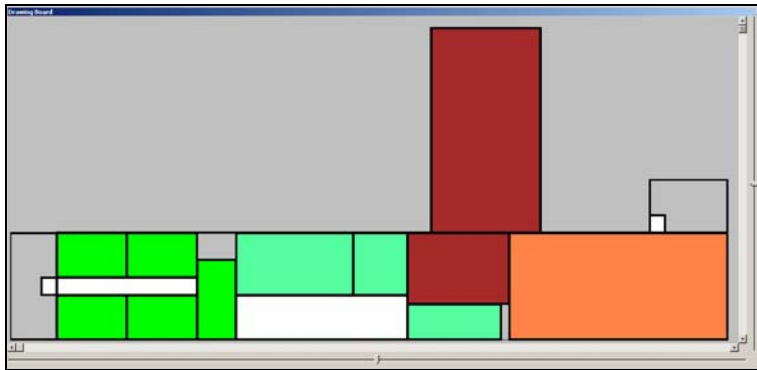


Abb. 7.31 Ergebnis 3 der Flächenneuordnung, Eiermann-Bau Apolda, EG

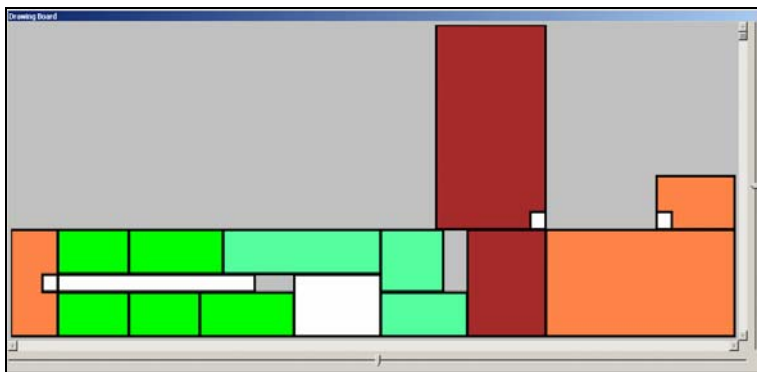


Abb. 7.32 Ergebnis 4 der Flächenneuordnung, Eiermann-Bau Apolda, EG

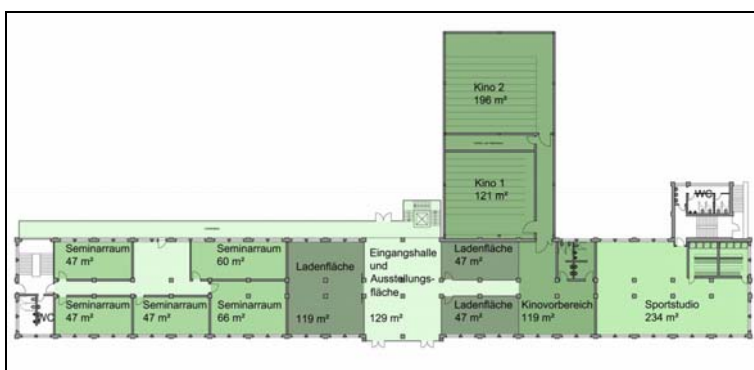


Abb. 7.33 Manuelle Entwurfslösung, Eiermann-Bau Apolda, EG [Marquardt und Raabe, 2002]

## Beispiel 2a

### Datei

7332\_2a.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.3)

### Zielfunktion

Gegeben ist ein  $900 \text{ m}^2$  großer rechteckiger Gebäudegrundriss mit den Seitenlängen 30 und 30 m. Finde innerhalb dieses Grundrisses eine Flächenanordnung für die 8 Flächen des Raumprogramms, so dass der Flächeninhalt (Variante 1) bzw. der Umfang (Variante 2) der Nutzflächen 6-8 ein Maximum erreicht und die nachfolgend genannten Nebenbedingungen erfüllt werden.

### Nebenbedingungen

Die Nebenbedingungen werden aus den Flächenerfordernissen des Flächenprogramms gebildet. Die Berechnung der Flächengrößen wird unter Verwendung Oberer und Unterer Schranken durchgeführt.

Nummer (min/max)	Raum	Farbe	Fläche	L und B
(Nr. 0)	Grundfläche	dunkelgrau	$900 \text{ m}^2$ (konstant)	30m / 30m
(Nr. 1)	Verkehrsfläche 1	blau	keine Beschränkung	5m / 20m
(Nr. 2)	Verkehrsfläche 2	blau	keine Beschränkung	5m / 20m
(Nr. 3)	Verkehrsfläche 3	blau	keine Beschränkung	5m / 20m
(Nr. 4)	Verkehrsfläche 4	blau	keine Beschränkung	5m / 20m
(Nr. 5)	Verkehrsfläche 5	blau	keine Beschränkung	5m / 20m
(Nr. 6)	Nutzfläche	hellgrau	$\geq 200 \text{ m}^2$	5m / 20m
(Nr. 7)	Nutzfläche	hellgrau	$\geq 200 \text{ m}^2$	5m / 20m
(Nr. 8)	Nutzfläche	hellgrau	$\geq 200 \text{ m}^2$	5m / 20m

Folgende Nebenbedingungen werden auf die Flächen des Raumprogramms angewendet:

- Alle Flächen befinden sich innerhalb des rechteckigen Gebäudegrundrisses (FInside).
- Alle Flächen müssen innerhalb eines 5m - Stützenrasters angeordnet werden.

- Keine der Flächen darf eine andere Fläche überschneiden (PInter).
- Die Gesamtfläche der Flächen entspricht der Grundfläche des Gebäudegrundrisses.
- Alle Flächen müssen eine Mindestbreite und Mindestlänge aufweisen, die eine dem Flächenprogramm entsprechende Nutzung zulassen (Length Constraint).
- Verbindungen zwischen Flächen müssen eine Mindestbreite von 5m aufweisen (Design Variablen).
- Jede Verkehrsfläche (blau) muss mindestens eine weitere Verkehrsfläche berühren.
- Alle Nutzflächen (hellgrau) müssen eine der 5 Verkehrsflächen berühren.

### Search Procedure

Es wird eine Search Procedure verwendet, welche die x- und y-Position aller Flächen auf ein Inkrement  $v$  beschränkt. Mit Hilfe dieses Inkrements wird ein fiktives 5m - Stützenraster abgebildet.

### Berechnungszeit

Numerische Lösung: 02,12 Sek. (Solver)

Graphische Lösung: 17,33 Sek. (Solver)

### Lösung

Für die programmiertechnische Umsetzung werden zwei neue Datenstrukturen verwendet, welche an die Programmiersprache C angelehnt sind. Mit Hilfe eines „Records“ wird eine Verbindung zwischen zwei zusammengehörigen Integer-Werten gebildet. Unter Verwendung von „Sets“ werden diese Records (hier die Nummern der Units 1-5) initialisiert.

Deklaration des Records:

```
struct connection
{
    int i;
    int j;
};
```

Deklaration des Sets:

```
{connection} fconnectionSet1={<1,2>,<1,3>,<1,4>,<1,5>};
```

Dieses Konstrukt erlaubt es, eine Nebenbedingung auf mehrere Units anzuwenden. Im obigen Fall werden die Verkehrsflächen 1-5 in einem „fconnectionSet“ deklariert, so dass die Anwendung einer Nebenbedingung (hier Force Connection), unter Verwendung eines logischen Operators auf alle (`forall`) oder mindestens eine (`or`) Unit angewendet werden kann.

Constraint-Deklaration:

```
forall (c in fconnectionSet1)
or (c in fconnectionSet1)
```

Auf diese Datenstrukturen wird einerseits an dieser Stelle verwiesen, da mit ihrer Hilfe der Programmcode erheblich reduziert werden kann. Andererseits ist es mit dieser Datenstruktur möglich, komplexe rechtwinklige Units zu konstruieren. Da insbesondere bei den Verkehrsflächen die Anforderung besteht, dass eine große Anzahl von Units Berührungskanten zu einer Verkehrsfläche aufweisen, kann eine rechteckige Verkehrsfläche nur in den seltensten Fällen diese Anforderung erfüllen. Der potentielle, architektonisch sinnvolle Lösungsraum eines Optimierungsproblems wäre bei Verwendung rechteckiger Flächen derart eingeschränkt, so dass die meisten komplexen Probleme nicht lösbar wären. Das gleiche Konstrukt ist selbstverständlich auf jede andere Unit anwendbar, so dass auf diese Weise eine wichtige Entkopplung vom Dogma der Beschränkung auf rechteckige Units gegeben ist. Die Abbildungen (Abb. 7.34 - Abb. 7.35) verdeutlichen die Funktionalität interner ConnectionSets, welche hier auf 5 Verkehrsflächen (blau) angewendet werden.

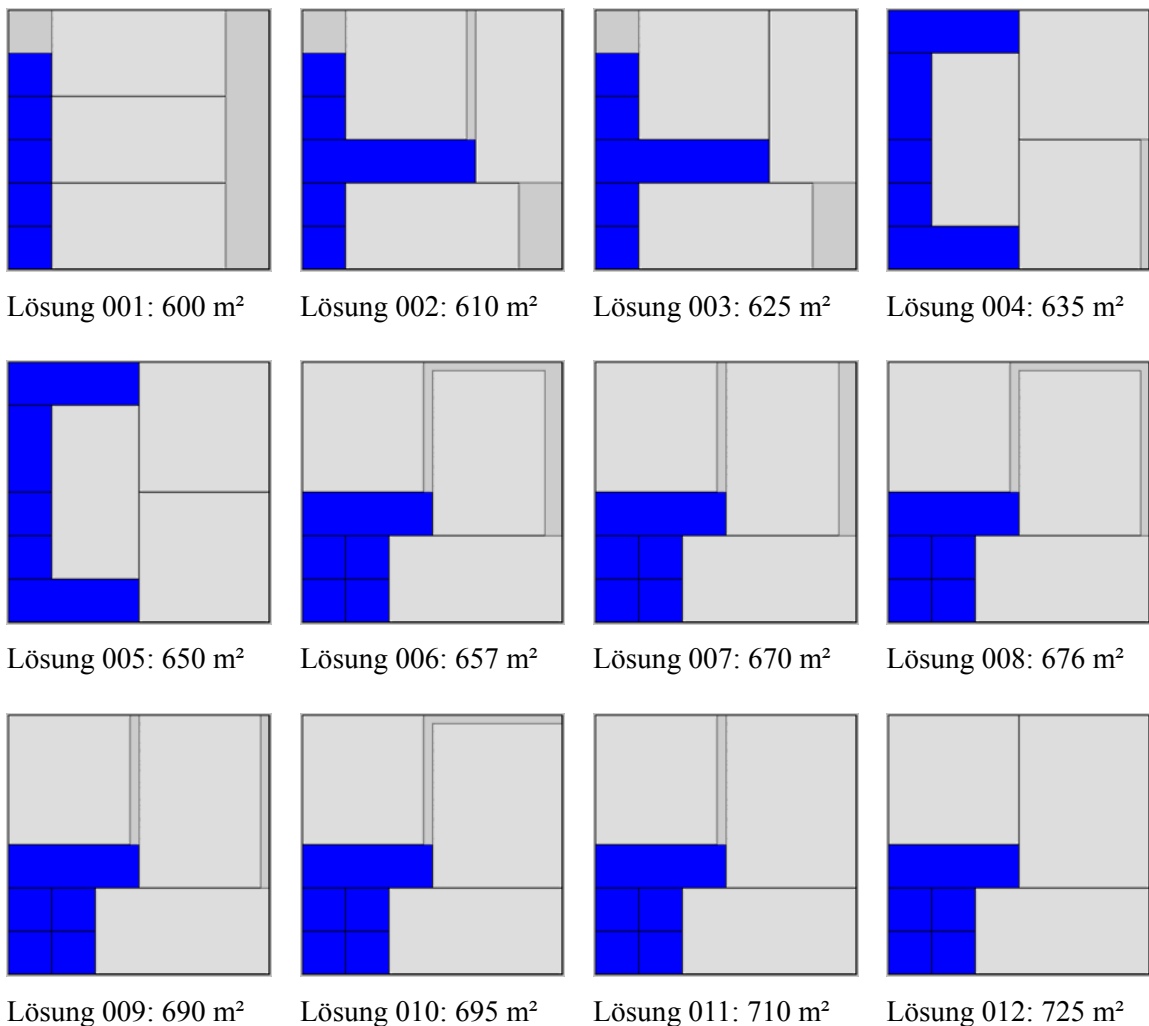


Abb. 7.34 Variante 1 - Maximierung des Flächeninhalts

Da die Verwendung quadratischer Zielfunktionen in Optimierungsberechnungen problematisch ist (vgl. Kap. 5.3.2.5), wird hier auch die Wirkungsweise der Flächenapproximation verdeutlicht. Das Endergebnis, d.h. die vollständige Erfüllung der Maximierungsfunktion entspricht in seiner architektonischen Ausformung exakt dem Ergebnis der Maximierung des Flächeninhaltes. Auch die Berechnungskosten sind mit jeweils 2,12 Sekunden identisch. Da die Berechnungskosten auf Grund quadratischer Zielfunktionen jedoch exponentiell ansteigen können und einige Algorithmen diese Funktionen überhaupt nicht unterstützen, ist die Übereinstimmung der Endergebnisse von besonderer Bedeutung.

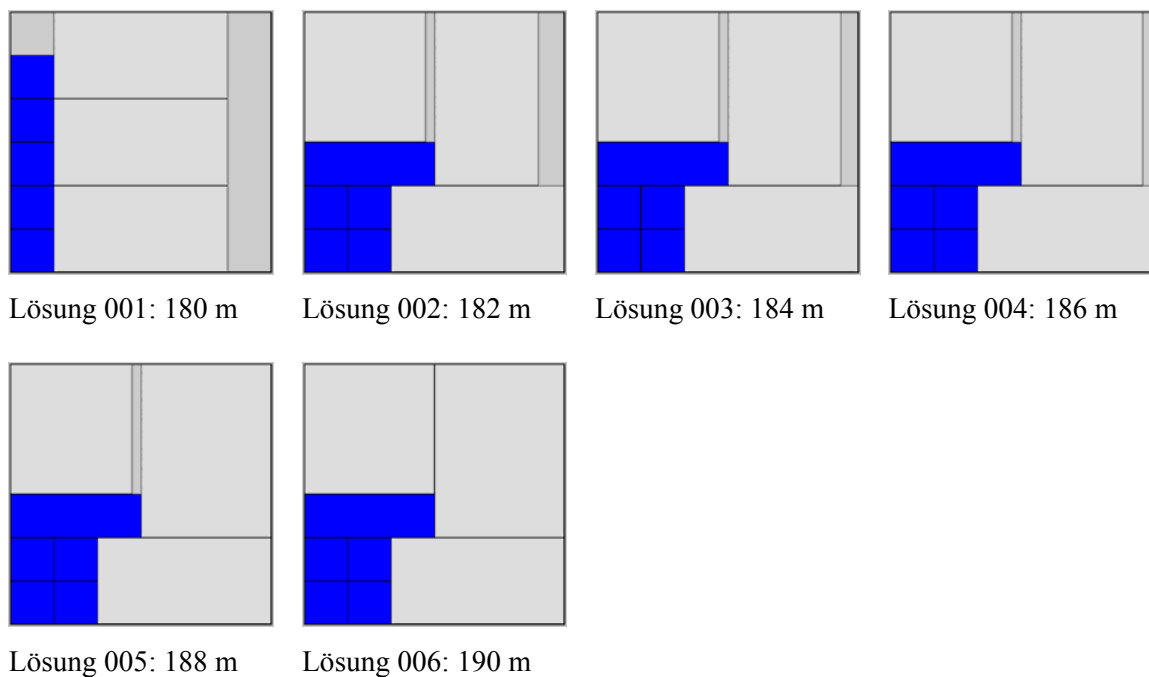


Abb. 7.35 Variante 2 - Maximierung des Umfangs (Flächenapproximation)

Zur Verdeutlichung der Verringerung der Berechnungskosten durch Search Procedures wurden Vergleichsrechnungen durchgeführt. In einer ersten Variante wurde ohne Search Procedure gearbeitet. In einer zweiten Variante wurde eine Search Procedure angewendet, welche ein fiktives Stützenraster von 5m auf die Verkehrsflächen 1-5 abbildet. Beide Berechnungen wurden nach jeweils über 24 Stunden Berechnungszeit ohne Ergebnis abgebrochen. Die zur Anwendung gelangte Search Procedure, die alle 8 Flächen auf ein fiktives Stützenraster von 5m beschränkt, benötigte hingegen lediglich eine Berechnungszeit von 02,12 Sekunden.

Das Ergebnis verdeutlicht einerseits, dass die Verwendung der in OPL programmierbaren Search Procedures außerordentlich effektiv sein kann. Andererseits veranschaulicht es auch die Schwierigkeit, eine Search Procedure so zu formulieren, dass sie effektiv auf die Lösung eines Optimierungsproblems einwirkt.



## Beispiel 2b

### Datei

7332\_2b.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.3)

### Zielfunktion

Es gilt die in Beispiel 2a (Kap. 7.3.3.2, Datei: 7332\_2a.mod) aufgeführte Zielfunktion (Flächeninhalt).

### Nebenbedingungen

Die Nebenbedingungen werden aus den Flächenerfordernissen des Raumprogramms gebildet. Die Berechnung der Flächengrößen wird unter Verwendung Oberer und Unterer Schranken durchgeführt.

Nummer (min/max)	Raum	Farbe	Fläche	L und B
(Nr. 0)	Grundfläche	dunkelgrau	900 m <sup>2</sup> (konstant)	30m / 30m
(Nr. 1)	Verkehrsfläche 1	blau	keine Beschränkung	3m / 30m
(Nr. 2)	Verkehrsfläche 2	blau	keine Beschränkung	3m / 30m
(Nr. 3)	Verkehrsfläche 3	blau	keine Beschränkung	3m / 30m
(Nr. 4)	Verkehrsfläche 4	blau	keine Beschränkung	3m / 30m
(Nr. 5)	Verkehrsfläche 5	blau	keine Beschränkung	3m / 30m
(Nr. 6)	Nutzfläche	hellgrau	$\geq 300 \text{ m}^2$	3m / 30m
(Nr. 7)	Nutzfläche	hellgrau	$\geq 150 \text{ m}^2$	3m / 30m
(Nr. 8)	Nutzfläche	hellgrau	$\geq 150 \text{ m}^2$	3m / 30m
(Nr. 9)	Eingangfläche	weiß	36 m <sup>2</sup> (konstant)	6m
(Nr. 10)	Eingangfläche	weiß	36 m <sup>2</sup> (konstant)	6m

Es gelten die in Beispiel 2a (Kap. 7.3.3.2, Datei: 7332\_2a.mod) aufgeführten Nebenbedingungen.

Verändert wird folgende Nebenbedingung:

- Verbindungen zwischen Flächen müssen eine Mindestbreite von 3m aufweisen (Design Variablen).

Zusätzlich werden folgende Nebenbedingungen eingeführt:

- Mindestens eine der Verkehrsflächen 1-5 muss eine die beiden Eingangsflächen 9 und 10 berühren.
- Position und Größe der Eingangsflächen sind konstant.

### Search Procedure

Variante 1: Es wird eine Search Procedure verwendet, welche die x- und y-Position aller Flächen auf ein Inkrement  $v$  beschränkt. Mit Hilfe dieses Inkrements wird ein fiktives 3m - Stützenraster abgebildet. Die x-Position der Flächen wird über ein absteigendes Inkrement gesucht.

Variante 2: Es werden 3 Search Procedures verwendet, die jeweils auf unterschiedliche Flächen angewendet werden. Da die Verkehrsfläche aus 5 einzelnen Flächen zusammengesetzt wird und da auf Grund der Verbindung der externen Flächen 9 und 10 mit der Verkehrsfläche, eine Lösung mit einer rechteckigen Verkehrsfläche unwahrscheinlich ist, werden für die Abszissen und Ordinaten der Referenzpunkte der Flächen 1 bis 3 und die Flächen 4 bis 5 unterschiedliche Search Procedures verwendet. Durch dieses Konstrukt wird der Suchraum effektiv begrenzt und eine schnelle Lösungsfindung ermöglicht. Diese Vorgehensweise schränkt den Lösungsraum nicht unzulässig ein, da bei Nicht-Existenz einer Lösung automatisch eine Default-Suchstrategie angewendet wird, die den gesamten potentiellen Lösungsraum durchsucht.

### Berechnungszeit

Numerische Lösung:	04,87 Sek. (Variante 1)
Graphische Lösung:	07,44 Sek. (Variante 1)
Numerische Lösung:	09,60 Sek. (Variante 2)
Graphische Lösung:	15,69 Sek. (Variante 2)

### Lösung

Beide Varianten berechnen innerhalb kürzester Zeit 20 Lösungen. Die Berechnungskosten nach 20 gefundenen Lösungen von Variante 1 liegen mit 04,87 Sekunden zwar unter denen von Variante 2 mit 09,60 Sekunden, die Qualität der Lösung von Variante 2 ist jedoch höher, da die Zielfunktion (Maximierung der Nutzfläche) in größerem Ausmaß erfüllt worden ist. Die Berechnung beider Varianten endet nicht nach 20 gefundenen Lösungen. Es schließt sich eine längere Berechnungszeit an, die für beide Varianten letztlich zum Endergebnis der maximal erzielbaren Nutzfläche führt. Diese Vorgehensweise des Solvers ist bei jeder durchgeführten Berechnung zu erkennen. Eine Maximierungsberechnung endet erst, wenn der Solver den Maximalwert gefunden hat und die Existenz größerer Werte ausgeschlossen worden ist. Dies führt mitunter dazu,

dass bereits nach kurzer Zeit (vgl. Variante 2) die perfekte Lösung gefunden wird, der Berechnungsvorgang allerdings noch weitere Kosten vereinnahmt. Für den errechneten Maximalwert von 783 m<sup>2</sup> Nutzfläche gibt es mindestens eine weitere architektonische Lösung. Bei einer Maximierungsberechnung werden allerdings nur Maxima und keine Varianten gesucht. Alternative Lösungen mit der gleichen Nutzfläche können jedoch in einer Berechnung ohne Zielfunktion (vgl. Kap. 5.3.1) problemlos gefunden werden. Die differenzierte Verwendung mehrerer Search Procedures hat klar erkennbare Vorteile gegenüber einer einzigen Search Procedure, welche auf alle Flächen gleichermaßen angewendet wird. Variante 2 findet das Optimum bereits nach wenigen Sekunden, wohingegen Variante 1 mehrere Stunden für die Lösungsfindung benötigt.

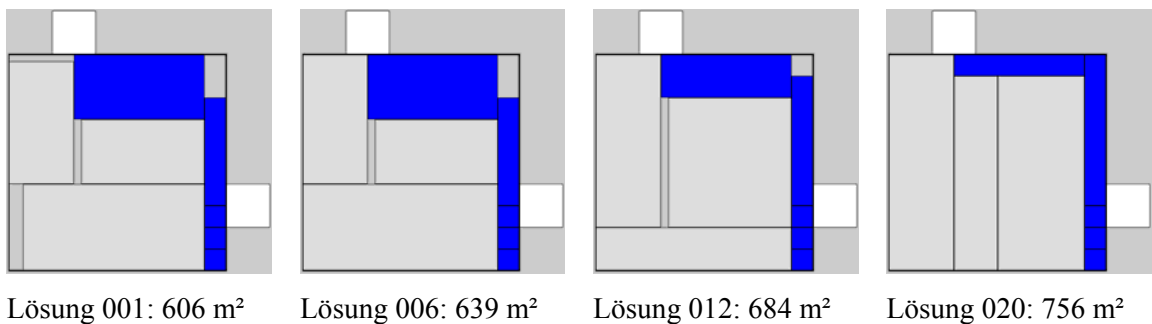


Abb. 7.36 Ergebnisse der Variante 1 unter Verwendung von einer Search Procedure

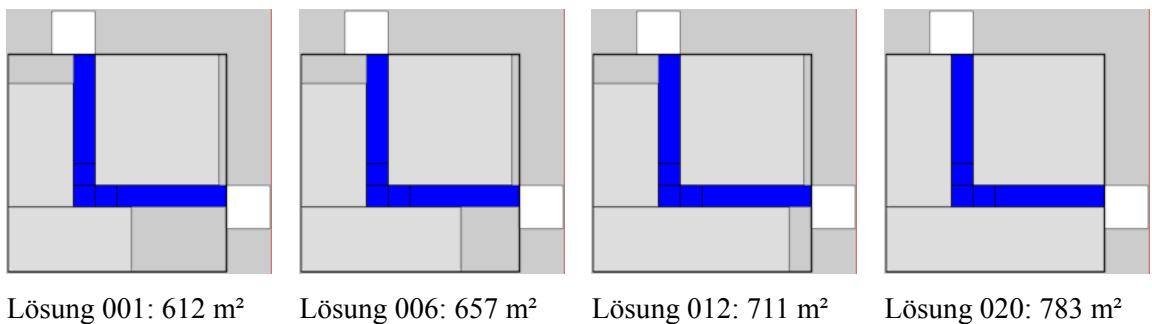


Abb. 7.37 Ergebnisse der Variante 2 unter Verwendung von drei Search Procedures

## Beispiel 2c

### Datei

7332\_2c.mod

### Mathematisches Modell

Mathematisches Modell 1 (vgl. Kap. 6.1.3)

### Zielfunktion

Gegeben ist ein 900 m<sup>2</sup> großer rechteckiger Gebäudegrundriss mit den Seitenlängen 30 und 30 m. Finde unter Erfüllung der nachfolgend genannten Nebenbedingungen innerhalb dieses Grundrisses eine Flächenanordnung für die 19 Flächen des Raumprogramms (Anm.: ...so dass, die Summe der Flächeninhalte der Flächen 1-8 der Gesamtfläche des Gebäudegrundrisses entspricht und die Flächen 11-19 innerhalb der Nutzfläche 6 angeordnet werden (Constraint Satisfaction)).

### Nebenbedingungen

Die Nebenbedingungen werden aus den Flächenerfordernissen des Raumprogramms gebildet. Die Berechnung der Flächengrößen wird unter Verwendung Oberer und Unterer Schranken durchgeführt.

Nummer	Raum	Farbe	Fläche	L und B (min/max)
(Nr. 0)	Grundfläche	dunkelgrau	900 m <sup>2</sup> (konstant)	30m / 30m
(Nr. 1)	Verkehrsfläche 1	blau	keine Beschränkung	3m / 30m
(Nr. 2)	Verkehrsfläche 2	blau	keine Beschränkung	3m / 30m
(Nr. 3)	Verkehrsfläche 3	blau	keine Beschränkung	3m / 30m
(Nr. 4)	Verkehrsfläche 4	blau	keine Beschränkung	3m / 30m
(Nr. 5)	Verkehrsfläche 5	blau	keine Beschränkung	3m / 30m
(Nr. 6)	Nutzfläche	hellgrau	≥ 300 m <sup>2</sup>	3m / 30m
(Nr. 7)	Nutzfläche	hellgrau	≥ 150 m <sup>2</sup>	3m / 30m
(Nr. 8)	Nutzfläche	hellgrau	≥ 150 m <sup>2</sup>	3m / 30m
(Nr. 9)	Eingangsfläche	weiß	36 m <sup>2</sup> (konstant)	6m / 6m
(Nr. 10)	Eingangsfläche	weiß	36 m <sup>2</sup> (konstant)	6m / 6m
(Nr. 11)	Nutzfläche	olivgrün	27 m <sup>2</sup>	3m / 30m
(Nr. 12)	Nutzfläche	olivgrün	27 m <sup>2</sup>	3m / 30m

(Nr. 13)	Nutzfläche	olivgrün	27 m <sup>2</sup>	3m / 30m
(Nr. 14)	Nutzfläche	sienna1	36 m <sup>2</sup>	3m / 30m
(Nr. 15)	Nutzfläche	sienna1	36 m <sup>2</sup>	3m / 30m
(Nr. 16)	Nutzfläche	sienna1	36 m <sup>2</sup>	3m / 30m
(Nr. 17)	Nutzfläche	waldgrün	54 m <sup>2</sup>	3m / 30m
(Nr. 18)	Nutzfläche	limonengrün	9 - 81 m <sup>2</sup>	3m / 30m
(Nr. 19)	Verkehrsfläche	rot	keine Beschränkung	3m / 30m

Es gelten die in Beispiel 2b (Kap. 7.3.3.2, Datei: 7332\_2b.mod) aufgeführten Nebenbedingungen.

Zusätzlich werden folgende Nebenbedingungen eingeführt:

- Die Summe der Flächeninhalte der Flächen 1-8 entspricht der Gesamtfläche des Gebäudegrundrisses.
- Die Flächen 11-19 befinden sich innerhalb der Fläche 6.
- Die Summe der Flächeninhalte der Flächen 11-19 entspricht der Gesamtfläche der Fläche 6.
- Die Verkehrsfläche 19 muss alle Nutzflächen 11-18 berühren.
- Die Verkehrsfläche 19 muss eine der Verkehrsflächen 1-5 berühren (Variante zur Beschränkung der Lösungsanzahl).

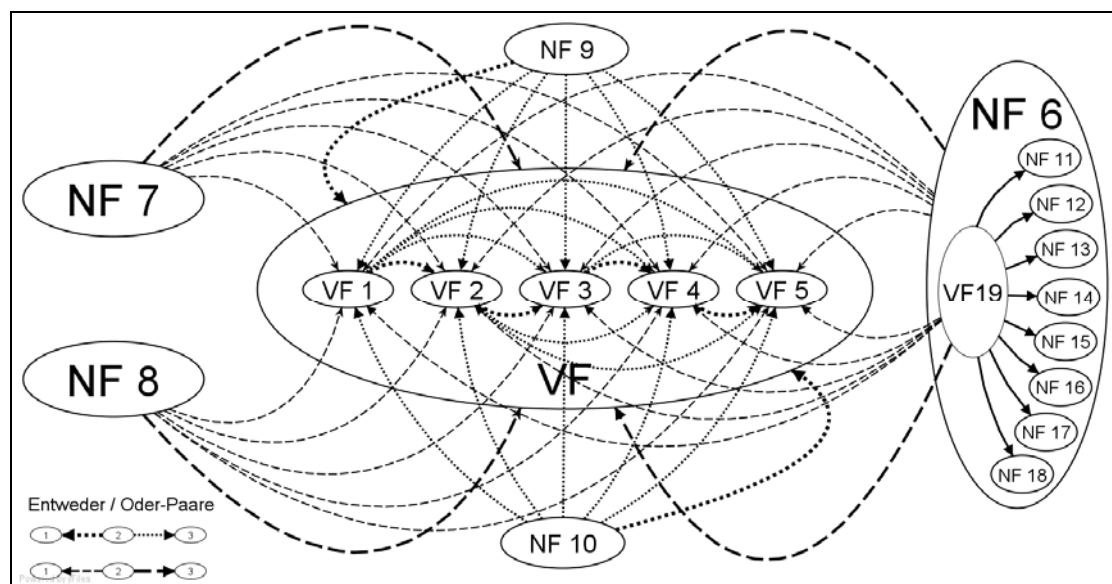


Abb. 7.38 Komplexität des Raumprogramms (Model: 7332\_2c.mod)

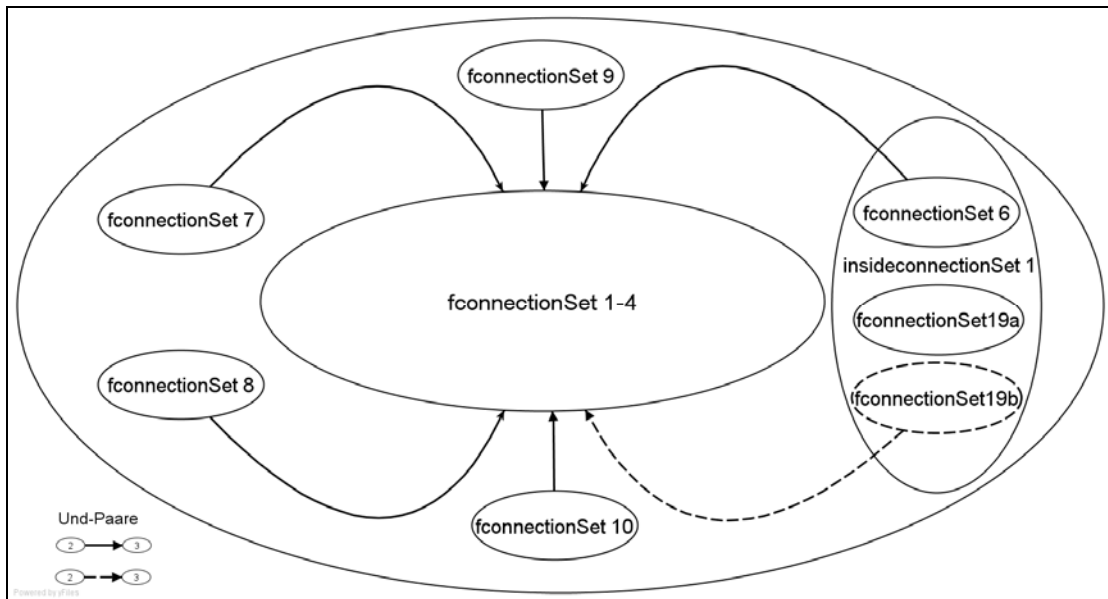


Abb. 7.39 Reduktion der Komplexität durch Verwendung von ConnectionSets

### Search Procedure

Es wird Variante 2 der Search Procedure aus Beispiel 2b (Kap. 7.3.3.2, Datei: 7332\_2b.mod) verwendet. Da die Berechnungsergebnisse unter Verwendung dieser Search Procedure zu Ergebnissen führen, die noch mit relativ hohen Berechnungskosten verbunden sind, wird eine vierte Search Procedure eingeführt, welche die Veränderung der Koordinaten der Referenzpunkte der Flächen 11-19 auf ein Inkrement von 3m beschränkt. Des Weiteren wird ein untergeordnetes Optimierungsziel - die Summe der Flächeninhalte der Flächen 11-19 muss der Gesamtfläche der Fläche 6 entsprechen - als Nebenbedingung definiert (Constraint Satisfaction). Diese Vorgehensweise sichert die Erfüllung aller als Nebenbedingung definierten Zielfunktionen und die Berechnung von Alternativen innerhalb einer Berechnung ohne Zielfunktion (vgl. Kap. 5.3.1). Zur weiteren Reduktion des Suchraumes kann mit Hilfe des „fconnectionSet19b“ eine Bedingung eingeführt werden, die ausschließlich Lösungen zulässt, bei denen die Berührung der Verkehrsfläche 19 (rot) mit einer der Verkehrsflächen 1-5 (blau) sichergestellt ist (Abb. 7.42).

### Berechnungszeit

Numerische Lösung: 66.684,46 Sek.

Graphische Lösung: -

### Lösung

Die nachfolgende Abbildung zeigt ein architektonisch noch nicht zufrieden stellendes Zwischenergebnis der Berechnung (Abb. 7.40). Es ist deutlich erkennbar, dass der Flächeninhalt der Verkehrsflächen 1 – 5 (blau) noch reduziert werden kann. Eine mögliche Vorgehensweise zur

Verbesserung des Ergebnisses wäre die Verwendung des Maximierungsergebnisses (783 m<sup>2</sup>) aus Beispiel 2b als Nebenbedingung. Diese Konstellation würde Lösungen ermitteln, welche die maximale Nutzfläche (Flächen 6-8), respektive die minimale Verkehrsfläche (Flächen 1-5) aufweisen.

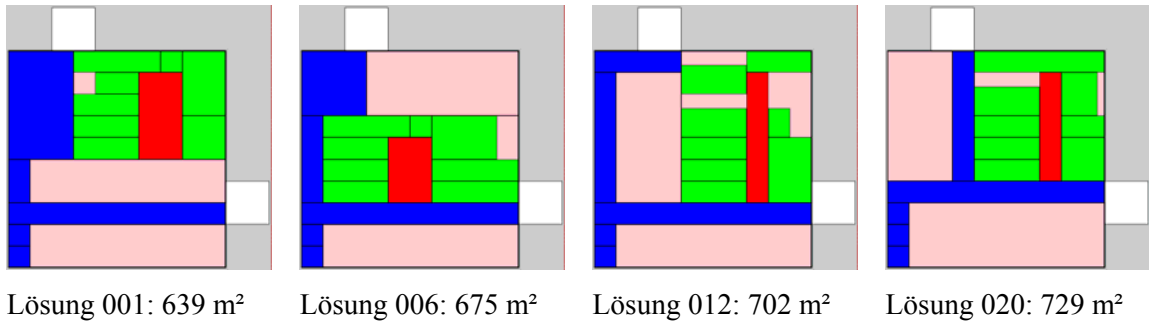


Abb. 7.40 Berechnungsergebnisse mit Search Procedure aus Variante 2 (7332\_2b.mod)

Abb. 7.41 zeigt eine Alternative zur oben beschriebenen Vorgehensweise. Statt der Verwendung des Optimierungsergebnisses aus Beispiel 2b wird hier eine zusätzliche Search Procedure eingeführt, welche die Veränderung der Referenzpunktkoordinaten inkrementell auf einen bestimmten Wertebereich beschränkt. Innerhalb von 18,52 Stunden werden 3,36 Milliarden Iterationen berechnet und 2.844 Lösungen erzeugt.

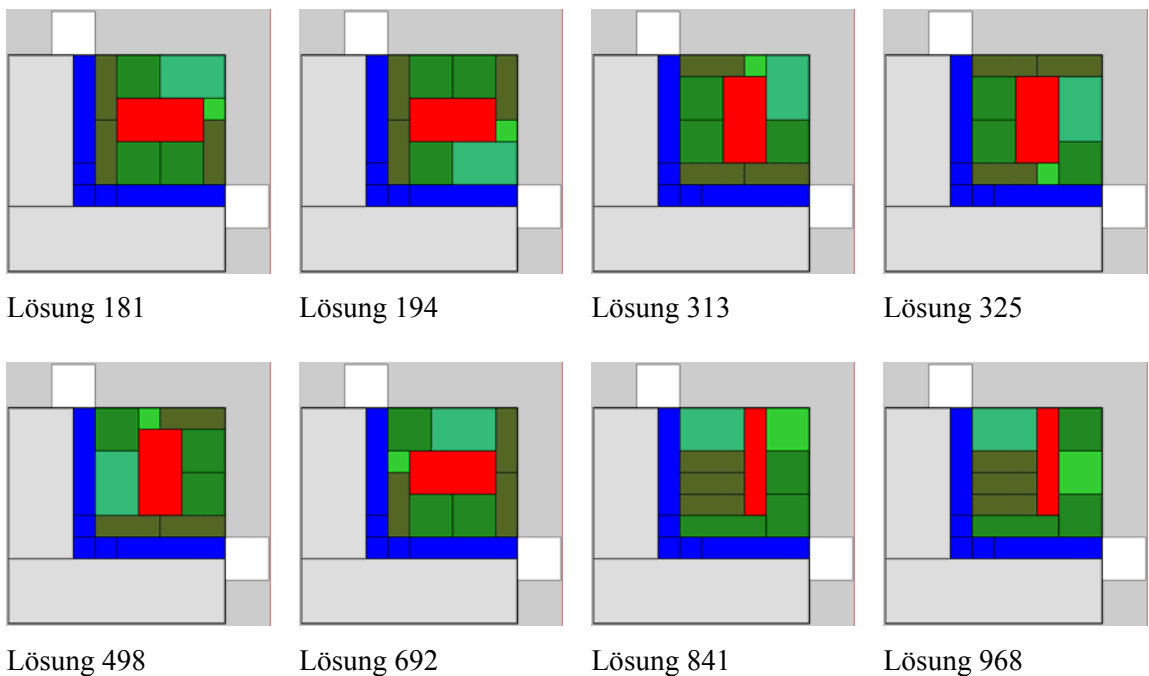


Abb. 7.41 Exemplarische Berechnungsergebnisse ohne fconnectionSet19b

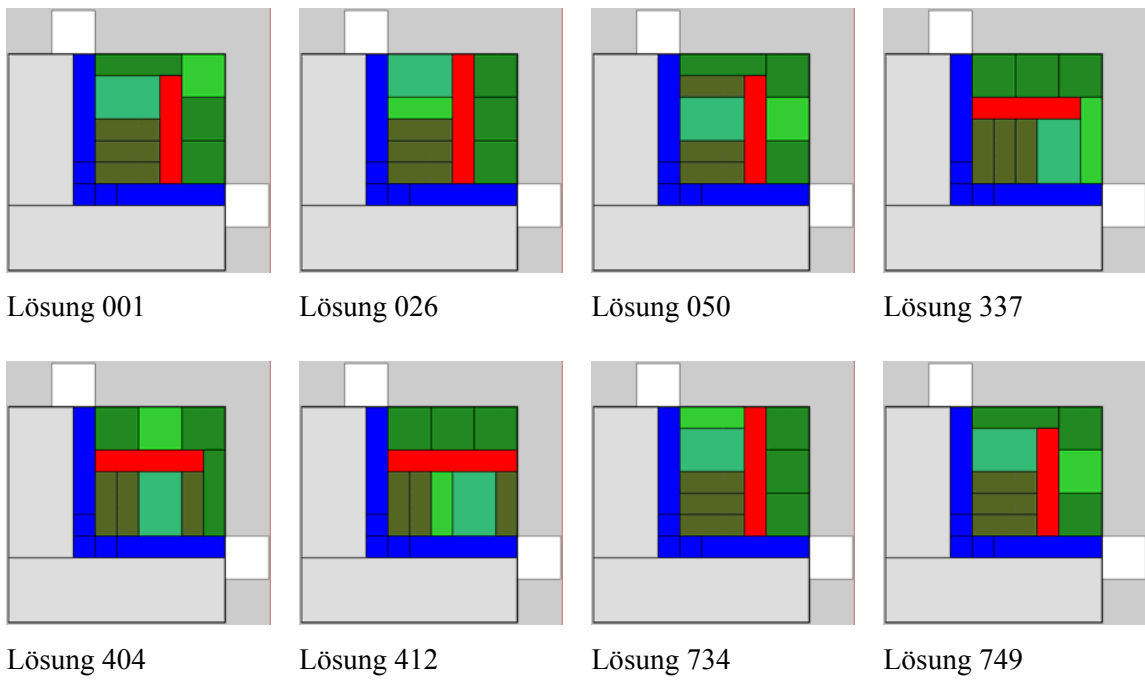


Abb. 7.42 Exemplarische Berechnungsergebnisse mit fconnectionSet19b

Eine umfassende Darstellung der Berechnungsergebnisse findet sich auf der beiliegenden CD.



### 7.3.4 Exemplarische Ergebnisse (Zerstörungsarmes Modell)

Das Zerstörungsarme Modell ist unabhängig von der Art der Grundrissanordnung. Rechteckige, rechtwinklige und nicht-rechtwinklige Gebäudegrundrisse und Units (vgl. Kap. 5.2.1) können gleichermaßen bearbeitet werden. Da rechteckige Units, durch ihre Beschränkung auf 4 parallele Kanten, hier nur eine Sonderform der rechtwinkligen Units darstellen, wird auf diese Unterscheidung nachfolgend nicht mehr eingegangen. Der Vorteil dieses Modells liegt jedoch insbesondere in der Anwendung auf komplexe Geometrien, wie sie vornehmlich im Gebäudebestand der Vorkriegszeit (vgl. Kap. 2) zu finden sind.

#### 7.3.4.1 Rechtwinklige und Nicht-Rechtwinklige Grundrisse und Units

Die Beispiele verwenden das mathematische Modell aus Kapitel 6.2.1.

Beispiel 1 zeigt die Anwendung des Prototyps auf eine Grundrisskonstellation mittlerer Komplexität mit 21 Räumen. Innerhalb des Grundrisses müssen alle potentiellen Anordnungsmöglichkeiten für vier Raumgruppen gefunden werden. Die gesuchten Raumgruppen müssen mindestens die Gruppengrößen aufweisen, die sich aus der Summe der Einzelflächen der Räume des Raumprogramms ergeben.

Beispiel 2a sucht innerhalb eines Grundrisses mittlerer Komplexität mit 24 Räumen vier Raumgruppen, deren Größe exakt der gemäß Raumprogramm definierten jeweiligen Gruppengröße entspricht. Beispiel 2b integriert eine Untere Schranke in eine veränderte Konstellation der Raumgruppengrößen von Beispiel 2a. Es ist bekannt, dass für die gesuchte Konstellation keine Lösung existiert. Die Verwendung der Schranke ermöglicht es, Lösungen zu erzeugen, die sich einer nicht existenten perfekten Lösung annähern. Diese Vorgehensweise besitzt besondere Relevanz für die Planungspraxis, da davon auszugehen ist, dass die perfekte Abbildung des geforderten Raumprogramms nur äußerst selten in einer existenten Grundrisskonstellation möglich ist.

Die Beispiele 3a – 3b verdeutlichen die Anwendung des Prototyps auf einen hochkomplexen Grundriss mit 78 Räumen. In Beispiel 3a wird innerhalb dieses Grundrisses eine aus 8 zusammenhängenden Räumen bestehende Raumgruppe gesucht. Für die Qualität der gefundenen Lösungen wird ein Gütemaß eingeführt, welches unter Einhaltung der Gesamtgruppengröße die Abweichung der gefundenen einzelnen Raumgrößen von den Raumgrößen des Raumprogramms bewertet. Als Ergebnis der Optimierungsberechnung werden 46 Lösungen ermittelt. Beispiel 3b vergrößert den Lösungsraum, indem es Obere und Untere Schranken mit einer prozentualen Abweichung von 3% für die Gesamtgruppengröße einführt. Im Ergebnis steht einer Verachtfachung der Lösungsanzahl lediglich eine Verdopplung der Berechnungskosten gegenüber.

Beispiel 4 sucht eine komplexe Gruppe mit 17 Räumen innerhalb der Grundrisskonstellation aus Beispiel 3a – 3b. Zur Beurteilung der erzielten 4935 Lösungen werden aus den Werten der Güte und der prozentualen Abweichung von der Gesamtgruppengröße durch Strafterme gewichtete Referenzwerte zur Beurteilung der Gesamtpformance eingeführt.

## Beispiel 1

### Datei

7341\_1.mod

### Mathematisches Modell

Mathematisches Modell 2 (vgl. Kap. 6.2.1)

### Zielfunktion

Gegeben ist ein Gebäudegrundriss mit einer Anzahl von 21 Räumen und einer Gesamtfläche von 290 m<sup>2</sup>. Finde innerhalb dieses Grundrisses 4 Raumgruppen (RG), mit einer Anzahl von 6, 6, 5 und 4 zusammenhängenden Räumen (R), deren jeweilige Raumgruppenfläche in Quadratmetern größer oder gleich der sich aus den Räumen des Raumprogramms ergebenden Raumgruppenflächen (RG1, ..., RG4;  $\sum = 285 \text{ m}^2$ ) ist.

### Nebenbedingungen

Raumgruppe	Raum 1	Raum 2	Raum 3	Raum 4	Raum 5	Raum 6
RG1 (90 m <sup>2</sup> )	25 m <sup>2</sup>	20 m <sup>2</sup>	15 m <sup>2</sup>	10 m <sup>2</sup>	10 m <sup>2</sup>	10 m <sup>2</sup>
RG2 (80 m <sup>2</sup> )	25 m <sup>2</sup>	20 m <sup>2</sup>	10 m <sup>2</sup>	10 m <sup>2</sup>	10 m <sup>2</sup>	5 m <sup>2</sup>
RG3 (70 m <sup>2</sup> )	20 m <sup>2</sup>	15 m <sup>2</sup>	15 m <sup>2</sup>	10 m <sup>2</sup>	10 m <sup>2</sup>	-
RG4 (45 m <sup>2</sup> )	15 m <sup>2</sup>	10 m <sup>2</sup>	10 m <sup>2</sup>	10 m <sup>2</sup>	-	-

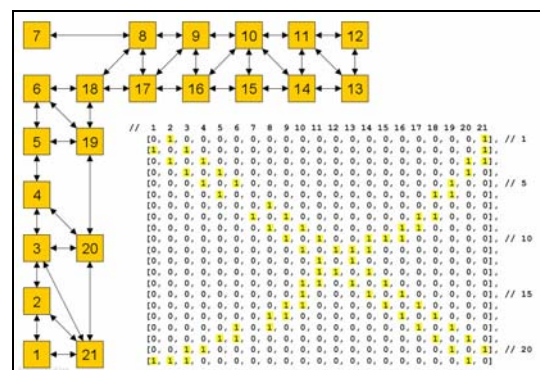
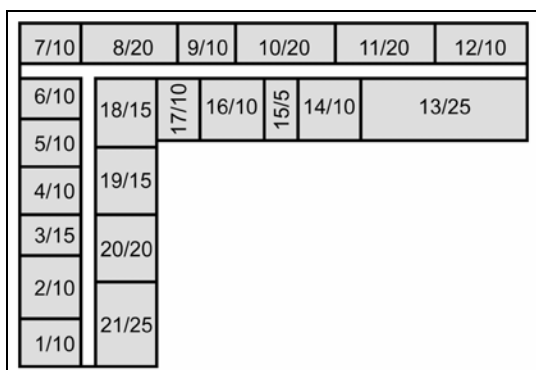


Abb. 7.43 Grundriss und Darstellung des resultierenden Graphen

**Berechnungszeit**

Numerische Lösung:	575,83 Sek. (Solver)
Graphische Lösung:	-
Anzahl der Ergebnisse:	8

**Lösung**

Der Prototyp sucht 4 Raumgruppen mit 6, 6, 5 und 4 zusammenhängenden Räumen sowie vorgegebenen Raumgruppengrößen RG1, ..., RG4. Die Größe der Raumgruppen ermittelt sich aus der Summe der Einzelflächen der Räume des Raumprogramms. Dabei werden nur die Gruppengrößen, nicht aber die einzelnen Raumgrößen gesucht. Es werden 8 Lösungen gefunden, welche die vorgenannten Anforderungen vollständig erfüllen. Die im Raumprogramm definierten Raumgrößen weichen nur geringfügig von den existenten Raumgrößen ab, so dass hier Lösungen erzielt werden können, bei welchen auch die Größen der einzelnen Räume innerhalb der gefundenen Gruppen in etwa den Vorgaben des Raumprogramms entsprechen. Dies muss jedoch nicht zwangsläufig bei anderen Raumprogrammen der Fall sein. Eine 70 m<sup>2</sup> große Fläche kann sich bspw. aus 5 Räumen mit 20, 15, 15, 10, 10 m<sup>2</sup> oder 50, 5, 5, 5, 5 m<sup>2</sup> zusammensetzen. In beiden Fällen wären die Nebenbedingungen bezüglich der Raumgruppengrößen und der Anzahl der Räume erfüllt. Architektonisch weisen diese Lösungen jedoch erhebliche Unterschiede auf, so dass die Suche nach Raumgruppengrößen in Kombination mit der Raumanzahl nur als erster Schritt zur Analyse einer gegebenen Grundrisskonstellation zu werten ist. Die zusätzliche Suche nach den Raumgrößen wird in Beispiel 3 verdeutlicht. Bei der Suche nach Raumgruppengrößen kann es Konstellationen geben, bei denen keine Lösung gefunden wird. Beispiel 2b zeigt die potentielle Vergrößerung des Lösungsraumes durch die Verwendung einer Unteren Schranke.

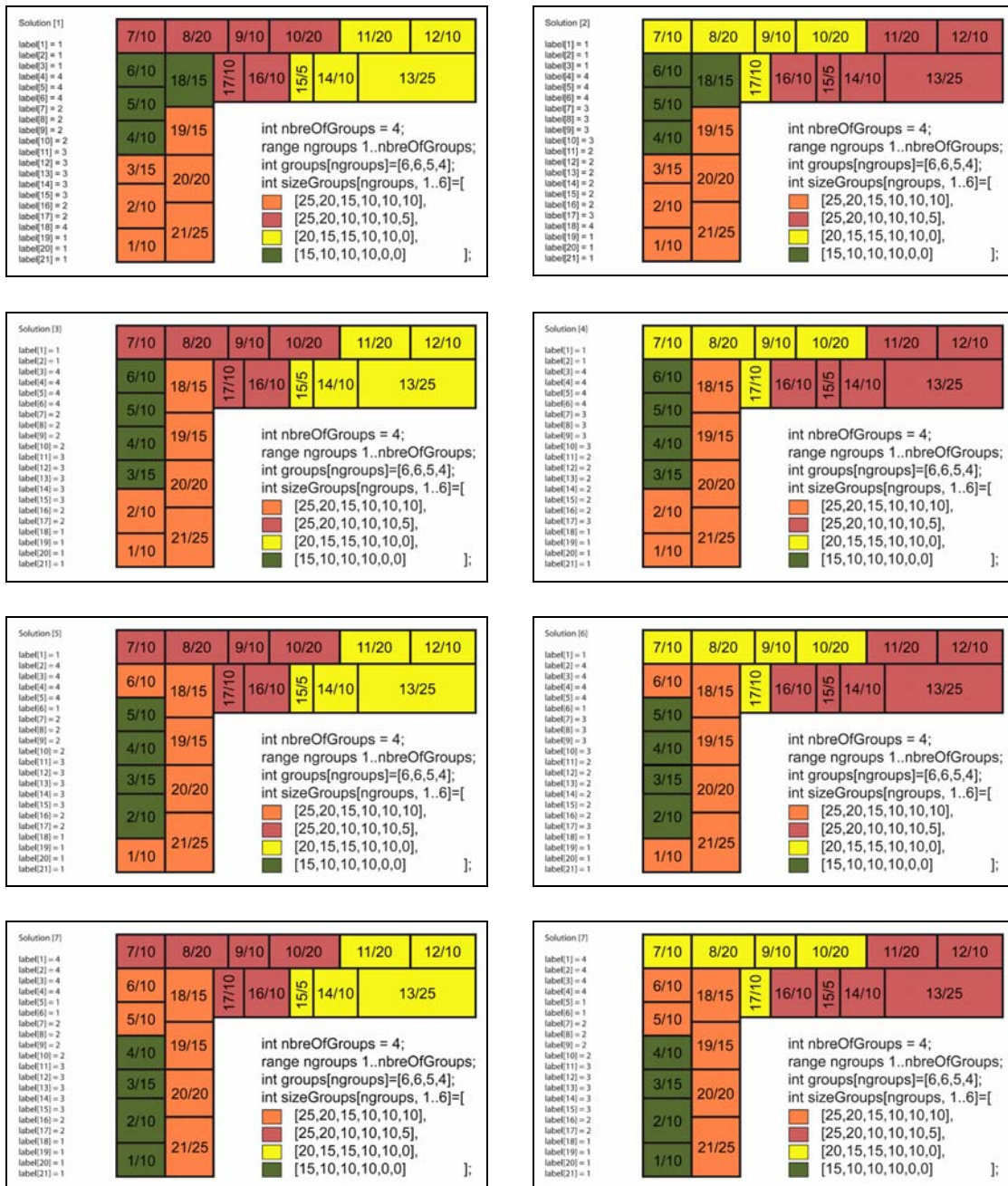


Abb. 7.44 Ergebnisse des Flächentauschs

## Beispiel 2a

### Datei

7341\_2.prj, 7341\_2.mod, 7341\_2.dat

### Mathematisches Modell

Mathematisches Modell 2 (vgl. Kap. 6.2.1)

### Zielfunktion

Gegeben ist ein Gebäudegrundriss mit einer Anzahl von 24 Räumen und einer Gesamtfläche von 369 m<sup>2</sup>. Finde innerhalb dieses Grundrisses 4 Raumgruppen (RG), mit einer Anzahl von 7, 6, 6 und 5 zusammenhängenden Räumen (R), deren jeweilige Raumgruppenfläche in Quadratmetern gleich der gem. Raumprogramm geforderten Raumgruppenflächen (RG1, ..., RG4;  $\sum = 369 \text{ m}^2$ ) ist.

### Nebenbedingungen

Raumgruppe	Raumgruppengröße	Anzahl der Räume
RG1	118 m <sup>2</sup>	7
RG2	73 m <sup>2</sup>	6
RG3	90 m <sup>2</sup>	6
RG4	88 m <sup>2</sup>	5
$\Sigma$	369 m <sup>2</sup>	24

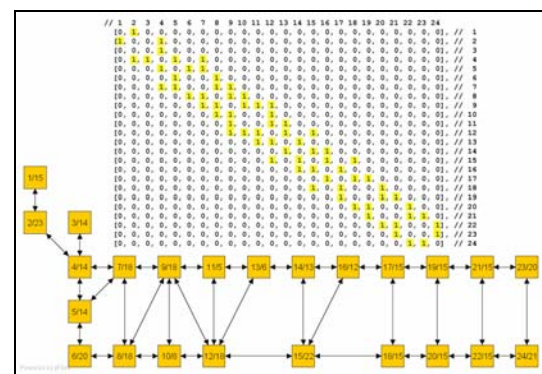
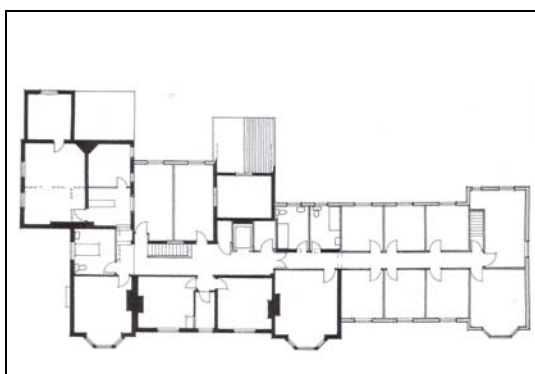


Abb. 7.45 Grundriss und Darstellung des resultierenden Graphen

### Berechnungszeit

Numerische Lösung: 6 Sek. (Solver)

Graphische Lösung: -

Anzahl der Ergebnisse: 2

### Lösung

Zur Verdeutlichung der späteren Verwendung einer Unteren Schranke (Beispiel 2b) werden hier (Beispiel 2a) Nebenbedingungen (Raumgruppengrößen und Anzahl der Räume) gewählt, welche unmittelbar aus der bestehenden Grundrisskonstellation extrahiert worden sind. Es ist daher mindestens eine Lösung zur Zielerfüllung bekannt. Erwartungsgemäß wird diese Lösung und eine weitere Lösung vom Solver ermittelt. Ferner wird in diesem Beispiel der Verzicht auf die Definition der Raumgrößen demonstriert. Diese Vorgehensweise ist dann sinnvoll anwendbar, wenn kein detailliertes Raumprogramm vorliegt, der Architekt jedoch in etwa die Größe einer gesuchten Gruppe und die Anzahl der Räume definieren kann.

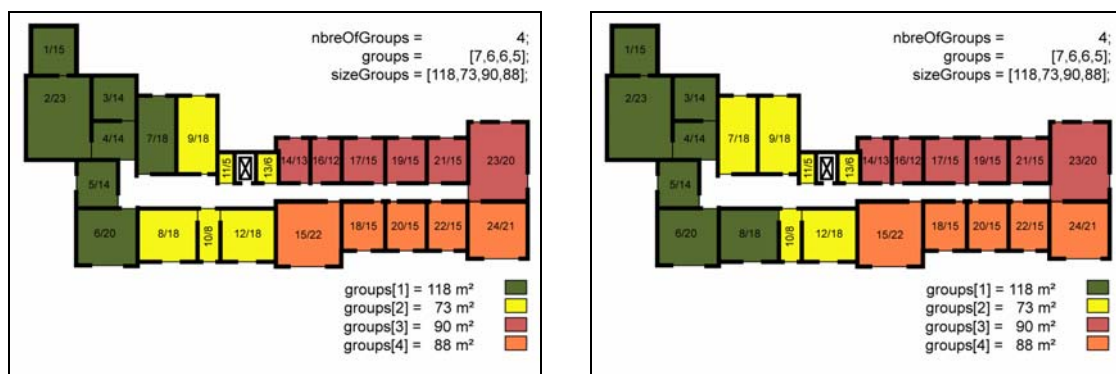


Abb. 7.46 Ergebnisse 1 und 2 des Flächentauschs

## Beispiel 2b

### Datei

7341\_2.prj, 7341\_2.mod, 7341\_2.dat

### Mathematisches Modell

Mathematisches Modell 2 (vgl. Kap. 6.2.1)

### Zielfunktion

Gegeben ist ein Gebäudegrundriss mit einer Anzahl von 24 Räumen und einer Gesamtfläche von 369 m<sup>2</sup>. Finde innerhalb dieses Grundrisses 4 Raumgruppen (RG), mit einer Anzahl von 7, 6, 6 und 5 zusammenhängenden Räumen (R), deren jeweilige Raumgruppenfläche in Quadratmetern gleich der gem. Raumprogramm geforderten Raumgruppenflächen (RG1, ..., RG4;  $\Sigma = 369 \text{ m}^2$ ) ist. Verwende bei Nicht-Existenz einer perfekten Lösung eine Untere Schranke.

### Nebenbedingungen

Raumgruppe	Raumgruppengröße	Anzahl der Räume
RG1	123 m <sup>2</sup>	7
RG2	68 m <sup>2</sup>	6
RG3	80 m <sup>2</sup>	6
RG4	98 m <sup>2</sup>	5
$\Sigma$	369 m <sup>2</sup>	24

### Berechnungszeit

Numerische Lösung: 94 Sek. (Solver)

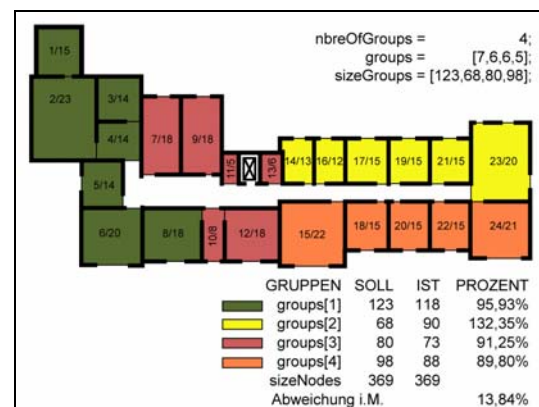
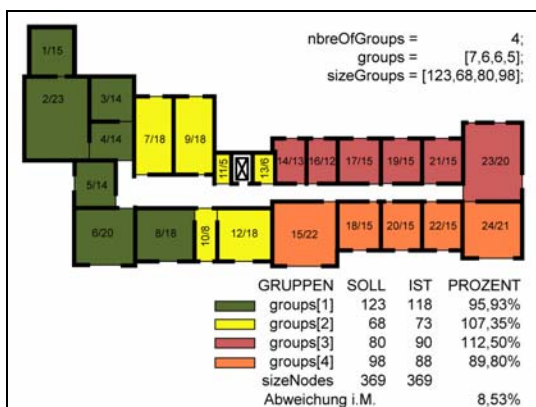
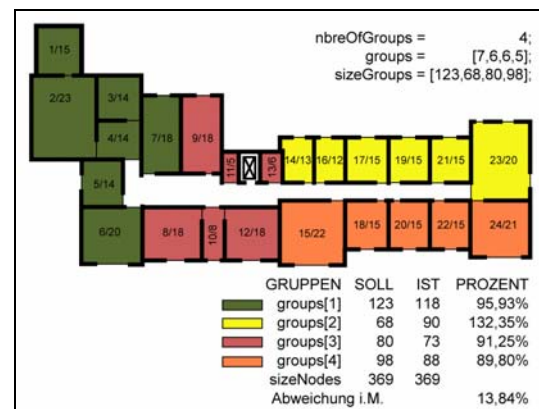
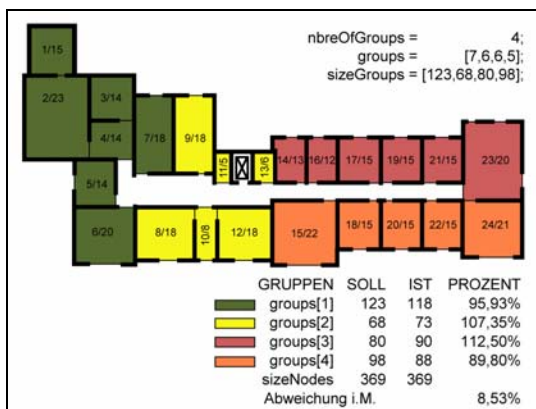
Graphische Lösung: -

Anzahl der Ergebnisse: 6

### Lösung

In Beispiel 2b wurden die Nebenbedingungen in Form der Raumgruppengrößen derart verändert, dass die Existenz einer Lösung nicht vorherbestimmt werden kann. Ein erster Berechnungsvorgang bestätigt, dass keine Lösung existiert, welche alle Nebenbedingungen erfüllt. Die Anwendung einer Unteren Schranke von 89% erzeugt 6 Lösungen innerhalb von 94 Sekunden. Diese Untere Schranke definiert die größte prozentuale Abweichung einer einzelnen Gruppe von der Zielfunktion (Raumgruppengröße). Die Gesamtgröße der Gruppen, d.h. die Aufteilung

der einzelnen Gruppen auf den gesamten Grundriss, wird durch die Verwendung Unterer Schranken in jedem Fall erfüllt. Da jedoch eine exakte Verteilung der existenten Räume auf die vorgegebenen Gruppengrößen in diesem Beispiel nicht möglich ist, versucht der Prototyp die Konstellation mit der geringsten Abweichung zu ermitteln. Dafür werden einige Gruppen mit größeren und andere mit kleineren Werten als vorgegeben belegt. Das Ergebnis ist eine Annäherung an die vom Benutzer definierten Werte. Da jedoch nicht ausschließlich die größte Abweichung einer einzelnen Raumgruppe entscheidend für die Wahl einer neuen Grundrisskonstellation ist, sind in diesem Beispiel auch die mittleren Abweichungen von der Zielfunktion berechnet worden. Diese Werte können dem Architekten als Anhaltspunkt für die zu bestimmende Performance einer Lösung dienen (vgl. Kap. 5.4). Zur Bewertung der Gesamtpformance zeigen die Beispiele 3a und b (Kap. 7.3.4.1) die prototypische Umsetzung zur Berechnung der „Güte“ einer Lösung.





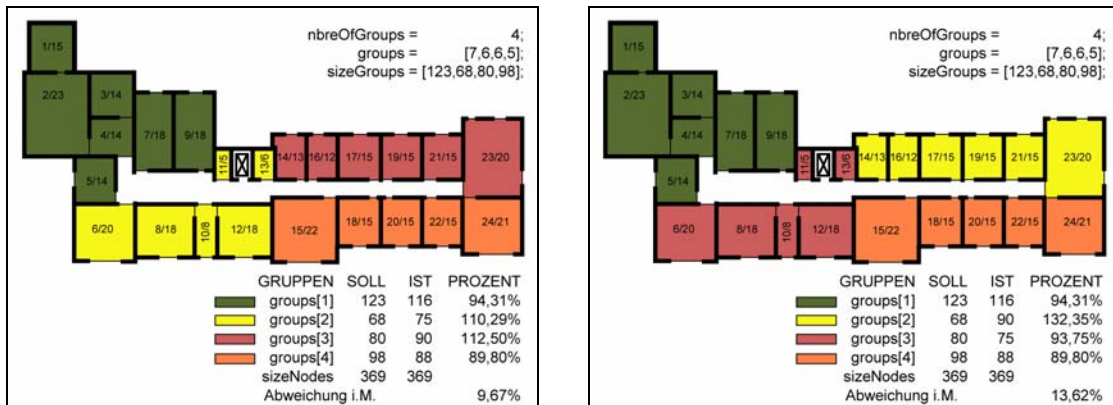


Abb. 7.47 Ergebnisse 1 bis 6 des Flächentauschs mit einer Unteren Schranke von 89 %

### Beispiel 3a

#### Datei

7341\_3.prj, 7341\_3.mod, 7341\_3.dat, 7341\_3.osc

#### Mathematisches Modell

Mathematisches Modell 2 (vgl. Kap. 6.2.1)

#### Zielfunktion

Gegeben ist ein Gebäudegrundriss mit einer Anzahl von 78 Räumen und einer Gesamtfläche von 1.956 m<sup>2</sup>. Finde innerhalb dieses Grundrisses eine Raumgruppe (RG), mit einer Anzahl von 8 zusammenhängenden Räumen (R), deren Raumgruppenfläche in Quadratmetern gleich der Summe der gem. Raumprogramm geforderten Einzelflächen (R1, ..., R8;  $\sum = 146 \text{ m}^2$ ) ist. Bestimme die Güte der ermittelten Lösungen.

#### Nebenbedingungen

Raumgruppe	Raum 1	Raum 2	Raum 3	Raum 4	Raum 5	Raum 6	Raum 7	Raum 8
1 (146 m <sup>2</sup> )	50 m <sup>2</sup>	30 m <sup>2</sup>	12 m <sup>2</sup>	12 m <sup>2</sup>	12 m <sup>2</sup>	12 m <sup>2</sup>	10 m <sup>2</sup>	8 m <sup>2</sup>

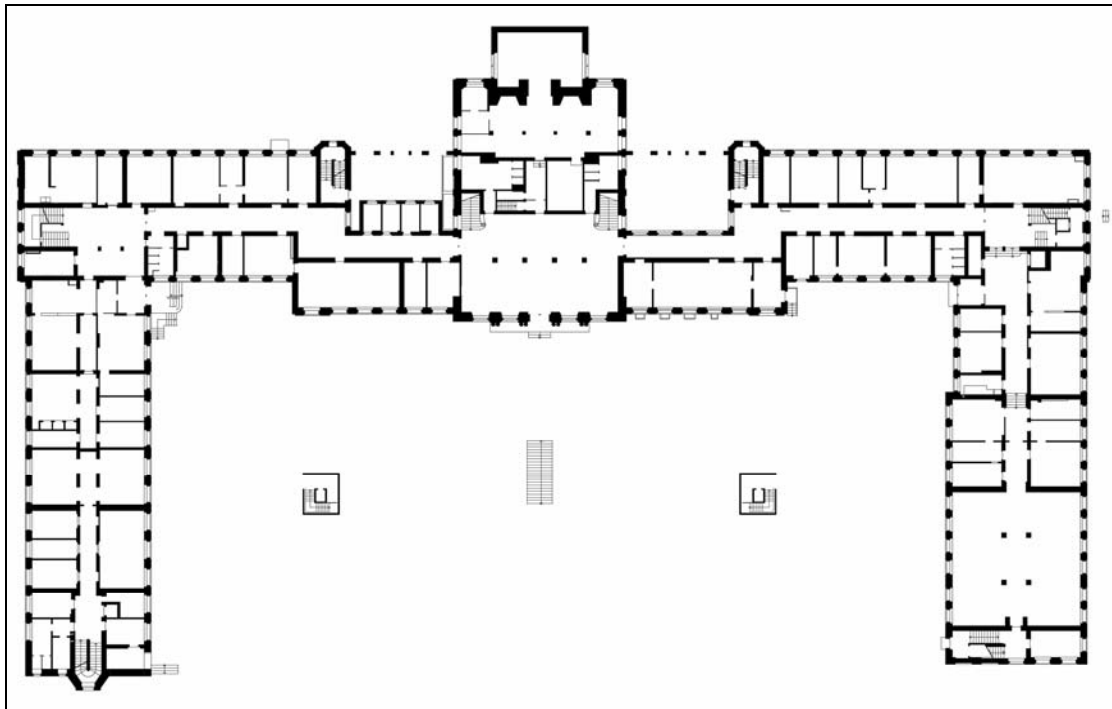


Abb. 7.48 Grundriss (Fritz-Foerster-Bau, Technische Universität Dresden)

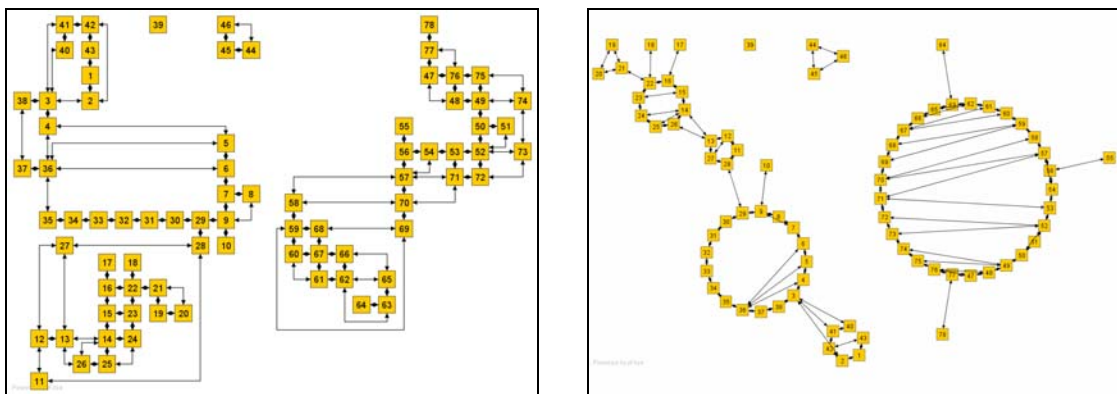


Abb. 7.49 Graphendarstellung

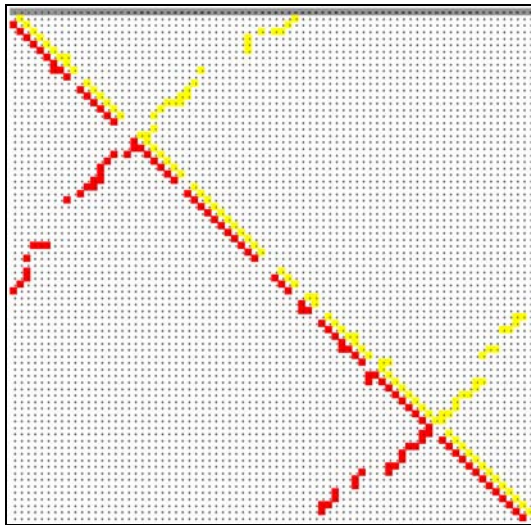


Abb. 7.50 Adjazenzmatrix

**Berechnungszeit**

Numerische Lösung: 845,01 Sek. (Solver)  
 Graphische Lösung: -  
 Anzahl der Ergebnisse: 46

**Lösung**

Der Prototyp sucht eine Raumgruppe mit 8 Räumen, vorgegebener Gruppengröße und vorgegebenen Raumgrößen innerhalb eines komplexen Grundrisses. Ohne die Verwendung Oberer und Unterer Schranken werden binnen 845 Sekunden insgesamt 46 Lösungen unterschiedlicher Güte gefunden. Alle Lösungen erfüllen die Anforderungen an die Gesamtgruppengröße (RG = 146 m<sup>2</sup>) und an die Anzahl (R = 8) der diese Gruppe bildenden Räume. Die Güte der Lösung determiniert die absolute Abweichung der gefundenen Raumgrößen von den gesuchten Raumgrößen (R) in Quadratmetern.

Beispiel: Gesucht wird eine Raumgruppe RG mit 3 Räumen (R1-R3) und einer Gesamtfläche von 60m<sup>2</sup>:

	RG (m <sup>2</sup> )	R1 (m <sup>2</sup> )	R2 (m <sup>2</sup> )	R3 (m <sup>2</sup> )	Güte (m <sup>2</sup> )
Gesucht werden:	60	40	10	10	0
Gefunden werden:	60	30	15	15	20
Gefunden werden:	60	20	20	20	40

Tab. 7.20 Ermittlung der Güte einer Lösung

Je geringer der Wert der Güte, d.h. je geringer die Abweichung der gefundenen Lösung von den gesuchten Raumgrößen, desto höher ist die Performance einer Lösung. Eine Güte von Null beschreibt eine Lösung, welche die Anforderungen an die Größe jedes einzelnen Raumes innerhalb der gesuchten Raumgruppe erfüllt.

	Raum 1	Raum 2	Raum 3	Raum 4	Raum 5	Raum 6	Raum 7	Raum 8	Summe	Güte (m <sup>2</sup> )	Annäherung (%)
Fläche Lösung 1	44	28	16	15	12	12	12	7	146	18	100,00
Fläche Lösung 2	44	27	21	13	12	12	10	7	146	20	100,00
Fläche Lösung 3	44	27	21	13	12	12	10	7	146	20	100,00
Fläche Lösung 4	44	28	23	12	12	11	9	7	146	22	100,00
Fläche Lösung 5	58	34	12	12	12	9	7	2	146	24	100,00
Fläche Lösung 6	57	24	18	12	11	10	8	6	146	26	100,00
Fläche Lösung 7	57	24	18	12	11	10	8	6	146	26	100,00
Fläche Lösung 8	57	24	18	12	11	10	8	6	146	26	100,00
Fläche Lösung 9	39	34	21	12	12	12	9	7	146	26	100,00
Fläche Lösung 10	39	34	21	12	12	12	9	7	146	26	100,00
Fläche Lösung 11	39	34	21	12	12	12	9	7	146	26	100,00
Fläche Lösung 12	39	34	21	12	12	12	9	7	146	26	100,00
sizeRooms	50	30	12	12	12	12	10	8	146	0	100,00

Tab. 7.21 Lösungen 1-12 sortiert nach Güte

Die nachfolgende Abbildung (Abb. 7.51) verdeutlicht die Ergebnisse des Flächentausches zur Erfüllung der Zielfunktion. Die Hofseite des Gebäudes mit den beiden Seitenflügeln ist nach Norden ausgerichtet. Die Lösungen 006-008 tauschen Räume im östlichen Gebäudeteil, während die Lösungen 001-005 und 009-012 Räume im westlichen Gebäudeteil tauschen.



Lösung 001, Güte 18



Lösung 002, Güte 20



Lösung 003, Güte 20



Lösung 004, Güte 22



Lösung 005, Güte 24



Lösung 006, Güte 26



Lösung 007, Güte 26



Lösung 008, Güte 26



Abb. 7.51 Lösungen 1-12 von 46

Die 46 ermittelten Lösungen variieren in der Güte der Lösung um Werte von 18 bis 52, bei gleichzeitiger Erfüllung aller übrigen Nebenbedingungen. Die Zielerfüllung der Nebenbedingungen und ein geringer Wert der Güte beschreiben eine gute Performance dieses Flächentausches. Es ist jedoch auch durchaus denkbar, dass eine noch bessere Lösung von der Erfüllung der übrigen Nebenbedingungen (Raumgruppengröße) abweicht und dadurch einen geringeren Wert der Güte, d.h. eine höhere Gesamt-Performance erzielt.

Beispiel: Gesucht wird eine Raumgruppe RG mit 3 Räumen (R1-R3) und einer Gesamtfläche von 60m<sup>2</sup>:

	RG (m <sup>2</sup> )	R1 (m <sup>2</sup> )	R2 (m <sup>2</sup> )	R3 (m <sup>2</sup> )	Güte (m <sup>2</sup> )	Abweichung (m <sup>2</sup> )
Gesucht werden:	60	40	10	10	0	0
Gefunden werden:	60	30	15	15	20	0
Gefunden werden:	62	42	8	12	6	2

Tab. 7.22 Ermittlung der Güte und der Abweichung einer Lösung

Beispiel 3b (Kap. 7.3.4.1) verdeutlicht diese Vorgehensweise durch die Verwendung Oberer und Unterer Schranken für die Bestimmung der Gesamtgruppengröße.

## Beispiel 3b

### Datei

7341\_3.prj, 7341\_3.mod, 7341\_3.dat, 7341\_3.osc

### Mathematisches Modell

Mathematisches Modell 2 (vgl. Kap. 6.2.1)

### Zielfunktion

Gegeben ist ein Gebäudegrundriss mit einer Anzahl von 78 Räumen und einer Gesamtfläche von 1.956 m<sup>2</sup>. Finde innerhalb dieses Grundrisses eine Raumgruppe (RG), mit einer Anzahl von 8 zusammenhängenden Räumen (R), deren Gesamtfläche in Quadratmetern gleich der Summe der gem. Raumprogramm geforderten Einzelflächen (R1, ..., R8;  $\sum = 146 \text{ m}^2$ ) ist. Bestimme die Güte der ermittelten Lösungen.

### Nebenbedingungen

Raumgruppe	Raum 1	Raum 2	Raum 3	Raum 4	Raum 5	Raum 6	Raum 7	Raum 8
1 (146 m <sup>2</sup> )	50 m <sup>2</sup>	30 m <sup>2</sup>	12 m <sup>2</sup>	12 m <sup>2</sup>	12 m <sup>2</sup>	12 m <sup>2</sup>	10 m <sup>2</sup>	8 m <sup>2</sup>

Es wird folgende Nebenbedingungen eingeführt:

- Die Größe der gesuchten Raumgruppe darf sich innerhalb einer Oberen und Unteren Schranke bewegen.
- Untere Schranke: 97 %
- Obere Schranke: 103 %

### Berechnungszeit

Numerische Lösung: 1.620,00 Sek. (Solver)

Graphische Lösung: -

Anzahl der Ergebnisse: 381

### Lösung

Durch die Verwendung einer Oberen und Unteren Schranke von jeweils ( $\pm$ ) 3% Abweichung von der Gesamtgruppengröße werden 381 Lösungen gefunden, deren Güte Werte zwischen 15 und 61 erreicht. Tab. 7.23 listet die 21 besten Lösungen sortiert nach Güte und spezifiziert deren

prozentuale Annäherung an die gesuchte Gesamtgruppengröße. Im Vergleich zu Beispiel 3a verbessert sich die höchste Güte (der geringste Wert) der Lösungen um 3 m<sup>2</sup>, bei einem Erfüllungsgrad der Zielfunktion von 99,32%, d.h. einer Abweichung von lediglich 1 m<sup>2</sup> von der gesuchten Gesamtgruppengröße. Dieses Ergebnis belegt, dass die Verwendung Oberer und Unterer Schranken sinnvoll ist, da der Lösungsraum erweitert und die Qualität der gefundenen Lösung verbessert werden kann. Abb. 7.52 verdeutlicht die Annäherung der einzelnen Lösungen an die gesuchten Raumgrößen.

	Raum 1	Raum 2	Raum 3	Raum 4	Raum 5	Raum 6	Raum 7	Raum 8	Summe	Güte (m <sup>2</sup> )	Annäherung (%)
Fläche Lösung 1	44	28	15	14	12	12	11	7	143	15	97,95
Fläche Lösung 2	44	34	15	12	12	12	9	7	145	15	99,32
Fläche Lösung 3	44	34	15	12	12	12	9	7	145	15	99,32
Fläche Lösung 4	44	28	15	14	12	12	12	7	144	16	98,63
Fläche Lösung 5	44	39	12	12	12	12	10	7	148	16	101,37
Fläche Lösung 6	44	28	16	15	12	12	9	7	143	17	97,95
Fläche Lösung 7	44	28	16	15	12	12	11	7	145	17	99,32
Fläche Lösung 8	44	39	12	12	12	12	9	7	147	17	100,68
Fläche Lösung 9	44	28	16	15	12	11	9	7	142	18	97,26
Fläche Lösung 10	44	28	16	15	12	12	12	7	146	18	100,00
Fläche Lösung 11	44	34	15	12	12	9	9	7	142	18	97,26
Fläche Lösung 12	44	34	15	12	12	9	9	7	142	18	97,26
Fläche Lösung 13	44	28	16	15	14	12	9	7	145	19	99,32
Fläche Lösung 14	44	28	16	15	14	12	11	9	149	19	102,05
Fläche Lösung 15	44	28	16	15	14	12	11	7	147	19	100,68
Fläche Lösung 16	44	28	16	15	14	12	9	7	145	19	99,32
Fläche Lösung 17	44	28	16	15	14	12	11	9	149	19	102,05
Fläche Lösung 18	44	28	16	15	14	12	11	7	147	19	100,68
Fläche Lösung 19	44	39	12	12	12	10	9	7	145	19	99,32
Fläche Lösung 20	44	27	21	12	12	12	10	7	145	19	99,32
Fläche Lösung 21	44	27	21	12	12	12	10	7	145	19	99,32
sizeRooms	50	30	12	12	12	12	10	8	146	0	100,00

Tab. 7.23 Lösungen 1-21 sortiert nach Güte



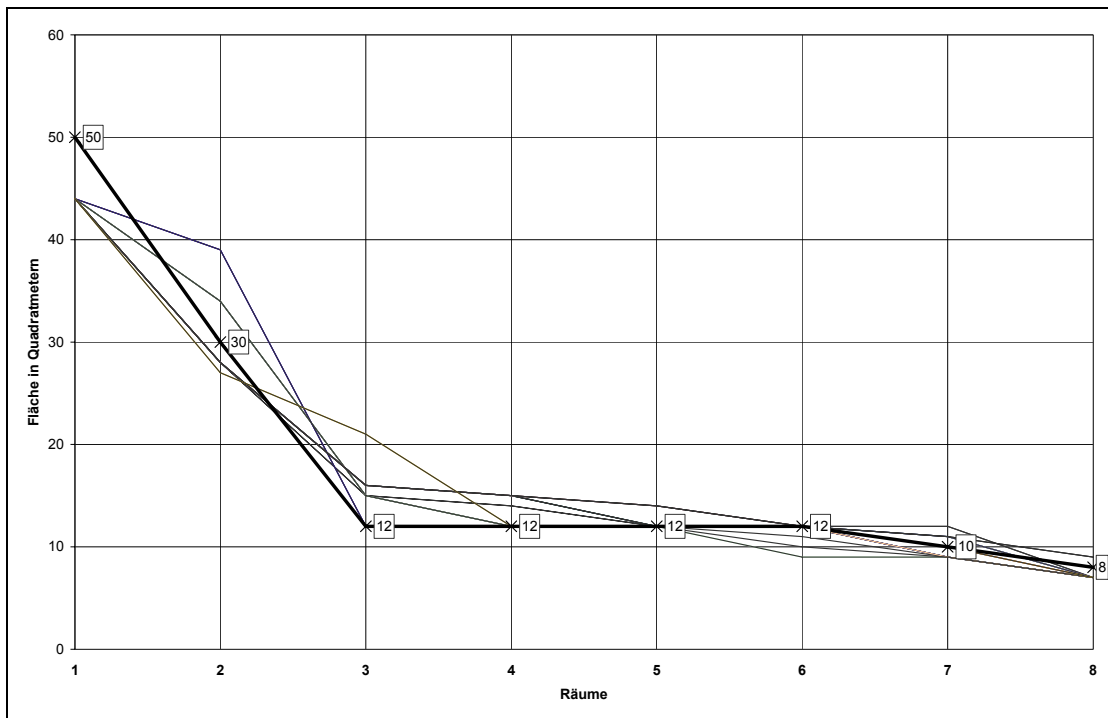
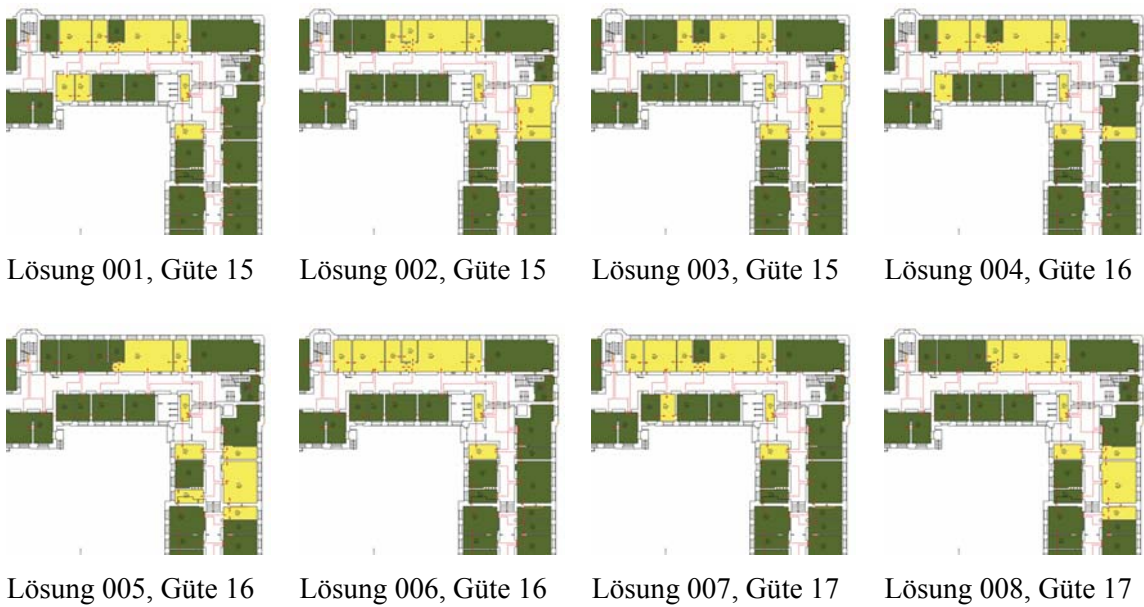


Abb. 7.52 Annäherung der Lösungen 1-21 an die Nebenbedingungen (sizeRooms)





Lösung 009, Güte 18



Lösung 010, Güte 18



Lösung 011, Güte 18



Lösung 012, Güte 18



Lösung 013, Güte 19



Lösung 014, Güte 19



Lösung 015, Güte 19



Lösung 016, Güte 19



Lösung 017, Güte 19



Lösung 018, Güte 19



Lösung 019, Güte 19



Lösung 020, Güte 19



Lösung 021, Güte 19

Abb. 7.53 Graphische Darstellung der Lösungen 1-21

Aus den ermittelten Werten der Güte einer Lösung und dem prozentualen Erfüllungsgrad der Zielfunktion lässt sich die Qualität einer Lösung noch nicht eindeutig ablesen. In Beispiel 4 wird daher, an einer komplexen Konstellation mit 17 zusammenhängenden Räumen, die Ermittlung der gewichteten Gesamtperformance durch die Verwendung von Straftermen demonstriert.

## Beispiel 4

### Datei

7341\_4.prj, 7341\_4.mod, 7341\_4.dat, 7341\_4.osc

### Mathematisches Modell

Mathematisches Modell 2 (vgl. Kap. 6.2.1)

### Zielfunktion

Gegeben ist ein Gebäudegrundriss mit einer Anzahl von 78 Räumen und einer Gesamtfläche von 1.956 m<sup>2</sup>. Finde innerhalb dieses Grundrisses eine Raumgruppe (RG), mit einer Anzahl von 17 zusammenhängenden Räumen (R), deren Gesamtfläche in Quadratmetern gleich der Summe der gem. Raumprogramm geforderten Einzelflächen (R1, ..., R17;  $\Sigma = 408 \text{ m}^2$ ) ist. Bestimme die Güte, die prozentuale Abweichung von der Gesamtgruppengröße und die gewichtete Gesamtperformance der ermittelten Lösungen.

### Nebenbedingungen

Raum (Nr.)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Größe (m <sup>2</sup> )	50	40	38	34	32	28	26	25	20	20	20	16	13	12	12	12	10
$\Sigma$ (m <sup>2</sup> )	408																

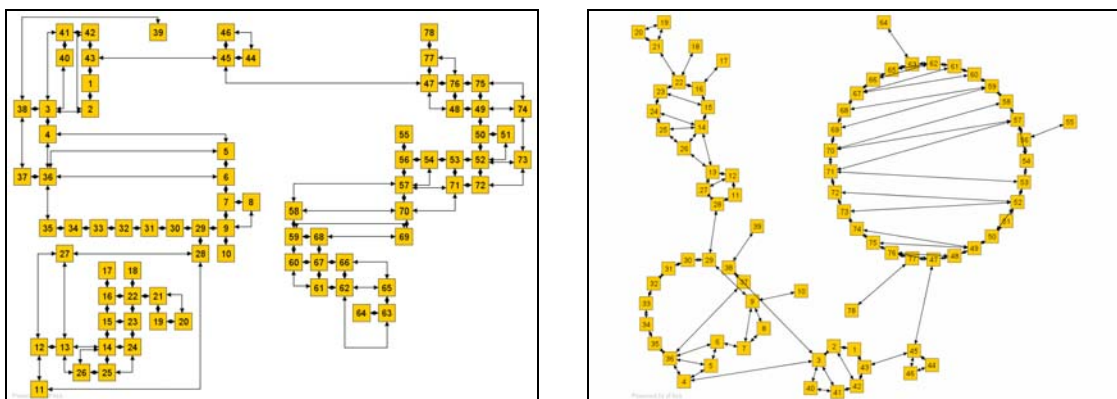


Abb. 7.54 Graphendarstellung

Es wird folgende Nebenbedingungen eingeführt:

- Die Größe der gesuchten Raumgruppe darf sich innerhalb einer Oberen und Unteren Schranke bewegen.
- Untere Schranke: 97 %
- Obere Schranke: 103 %

**Berechnungszeit**

Numerische Lösung: ca. 43.200,00 Sek. (Solver)  
 Graphische Lösung: -  
 Anzahl der Ergebnisse: 4935

**Lösung**

Innerhalb von ca. 12 Stunden werden 4935 Lösungen berechnet, deren Güte Werte zwischen 28 und 200 annimmt. Die Trendlinie des Berechnungsverlaufs (Abb. 7.55) verdeutlicht, dass die Erzeugung der Ergebnisse nicht linear ist, d.h. die Ergebnisse werden im zeitlichen Verlauf der Berechnung weder linear besser noch schlechter. Diese Erkenntnis hat zur Folge, dass bei dem verwendeten Algorithmus ein Berechnungsvorgang immer vollständig abgeschlossen sein muss, bevor ein aussagekräftiges Ergebnis vorliegt.

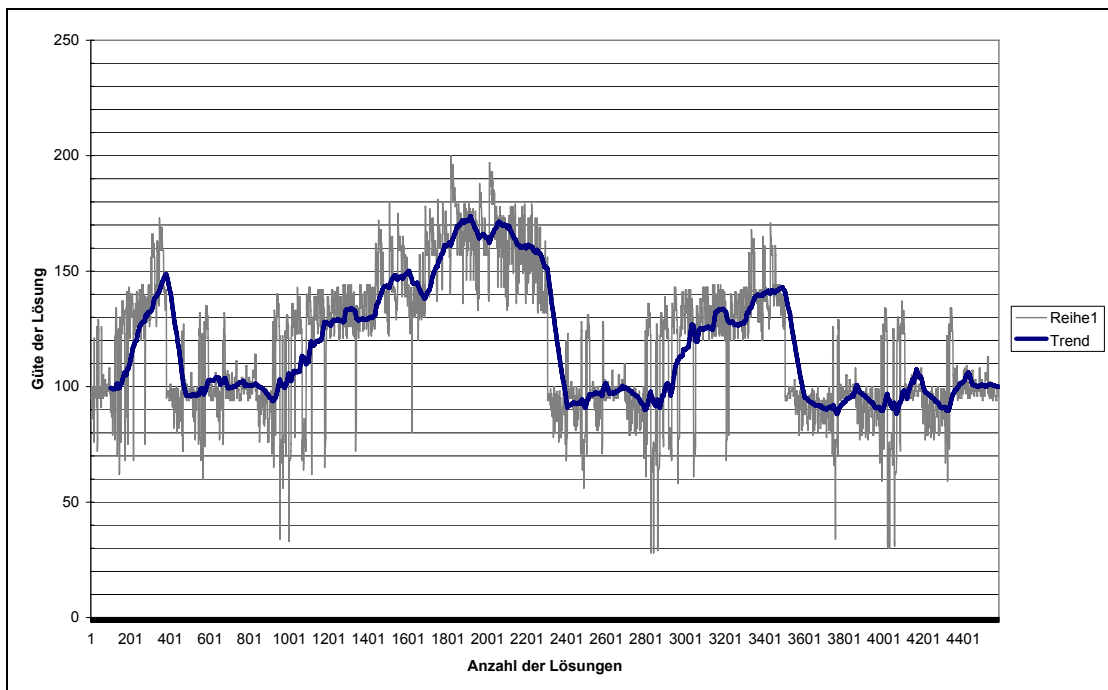


Abb. 7.55 Berechnungsverlauf und Trendlinie der Güte der Lösungen

In diesem Beispiel wurden 389.865 Ergebniszeilen erzeugt, welche die Variablenwerte, d.h. die Gruppenzugehörigkeit jedes einzelnen Raumes und die Güte der Lösung für das entsprechende Berechnungsergebnis dokumentieren, eine schnelle Auswertung der Ergebnisse jedoch erschweren. Dessen ungeachtet, gestattet die hohe Anzahl der Lösungen die graphische Auswertung der Häufigkeitsverteilung der Güte der Lösungen (Abb. 7.56). Dabei wird deutlich, dass nur eine geringe Anzahl von Lösungen eine Güte unter 40 m<sup>2</sup> erreicht und der weitaus größte Anteil der Lösungen eine Güte von 60-140 m<sup>2</sup> aufweist, deren architektonische Umsetzung keine akzeptablen Lösungen darstellt. Es sei jedoch angemerkt, dass dieses Performancemerkmal lediglich die Güte, d.h. die absolute Abweichung der Größe der einzelnen Räume untereinander, beschreibt. Alle 4935 erzielten Lösungen erfüllen die Zielfunktion hinsichtlich der Anzahl der gesuchten Räume und liegen innerhalb einer Abweichung von maximal 3 % (!) von der gesuchten Gesamtgruppengröße. Die Varianz der Ergebnisse zur Berechnung der Güte macht deutlich, dass es ebenfalls sinnvoll ist, Obere und Untere Schranken für deren Berechnung einzuführen. Tab. 7.24 und Abb. 7.57 zeigen die schlechtesten Lösungen, mit einer Güte von 197 bzw. 200 m<sup>2</sup> und einer Annäherung an die gesuchte Gesamtgruppengröße von 98,28 bzw. 97,55 %.

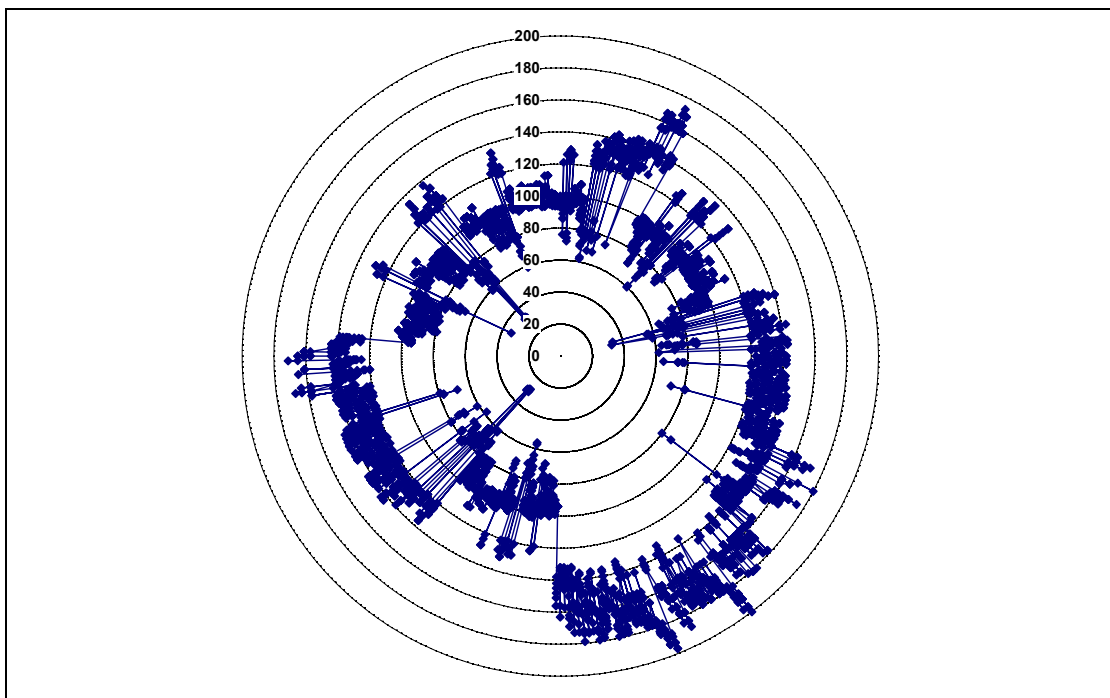


Abb. 7.56 Häufigkeitsverteilung der Güte der Lösungen

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	Summe	Güte (m <sup>2</sup> )	Ann. (%)	Abw. (%)
Fläche Lös. 1816	99	67	57	24	22	18	17	16	16	12	11	10	8	6	6	6	6	401	197	98,28	1,72
Fläche Lös. 2010	99	67	57	24	22	18	17	16	13	12	11	10	8	6	6	6	6	398	200	97,55	2,45
sizeRooms	50	40	38	34	32	28	26	25	20	20	20	16	13	12	12	12	10	408	0	100,00	0,00

Tab. 7.24 Schlechteste Lösungen Nr. 1816 und Nr. 2010 mit einer Güte von 197 und 200



Abb. 7.57 Graphische Darstellung der Lösungen Nr. 1816 und Nr. 2010

Zur Beurteilung der Lösung werden aus den Werten der Güte und der prozentualen Abweichung von der Gesamtgruppengröße gewichtete Referenzwerte gebildet, die Aufschluss über die Gesamtperformance der Lösungen geben (vgl. Kap. 5.4.). Aus den Werten der 16 besten Lösungen (Tab. 7.25 und Abb. 7.58) ist ersichtlich, dass die Lösungen 1 und 2 die beste Güte von 28 aufweisen, deren Abweichung von der Gesamtgruppengröße allerdings jeweils 2,45% beträgt. Die Lösungen 4 und 10 weisen eine schlechtere Güte von 29 und 31 auf, haben jedoch die mit 0,74% und 0,25% geringste Abweichung von der Gesamtgruppengröße (Abb. 7.59).

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	Summe	Güte (m <sup>2</sup> )	Ann. (%)	Abw. (%)
Fläche Lösung 1	57	42	33	31	31	28	26	24	18	18	17	15	13	12	12	11	10	398	28	97,55	2,45
Fläche Lösung 2	57	42	33	31	31	28	26	24	18	18	17	15	13	12	12	11	10	398	28	97,55	2,45
Fläche Lösung 3	57	42	33	31	31	28	25	24	18	18	17	15	13	12	12	11	10	397	29	97,30	2,70
Fläche Lösung 4	57	42	33	31	31	28	26	25	24	18	17	15	13	12	12	11	10	405	29	99,26	0,74
Fläche Lösung 5	57	42	33	31	31	28	26	24	18	18	17	15	15	12	12	11	10	400	30	98,04	1,96
Fläche Lösung 6	57	42	33	31	31	28	26	24	18	17	16	15	13	12	12	11	10	396	30	97,06	2,94
Fläche Lösung 7	57	42	33	31	31	28	26	24	18	18	17	15	14	13	12	11	10	400	30	98,04	1,96
Fläche Lösung 8	57	42	33	31	31	28	26	24	18	18	17	15	14	13	12	11	10	400	30	98,04	1,96
Fläche Lösung 9	57	42	33	31	31	28	25	24	18	18	17	15	14	13	12	11	10	399	31	97,79	2,21
Fläche Lösung 10	57	42	33	31	31	28	26	25	24	18	17	15	14	13	12	11	10	407	31	99,75	0,25
Fläche Lösung 11	57	42	33	31	31	28	26	24	18	18	17	15	15	14	12	11	10	402	32	98,53	1,47
Fläche Lösung 12	57	42	33	31	31	28	26	24	18	17	16	15	14	13	12	11	10	398	32	97,55	2,45
Fläche Lösung 13	57	42	33	31	31	28	26	25	24	18	17	15	13	12	11	10	8	401	33	98,28	1,72
Fläche Lösung 14	57	42	33	31	31	28	26	24	18	17	15	15	14	13	12	11	10	397	33	97,30	2,70
Fläche Lösung 15	57	42	33	31	31	28	26	24	18	18	17	15	15	12	11	10	8	396	34	97,06	2,94
Fläche Lösung 16	57	42	33	31	31	28	26	24	18	17	15	14	14	13	12	11	10	396	34	97,06	2,94
sizeRooms	50	40	38	34	32	28	26	25	20	20	20	16	13	12	12	12	10	408	0	100,00	0,00

Tab. 7.25 Lösungen 1-16 sortiert nach Güte

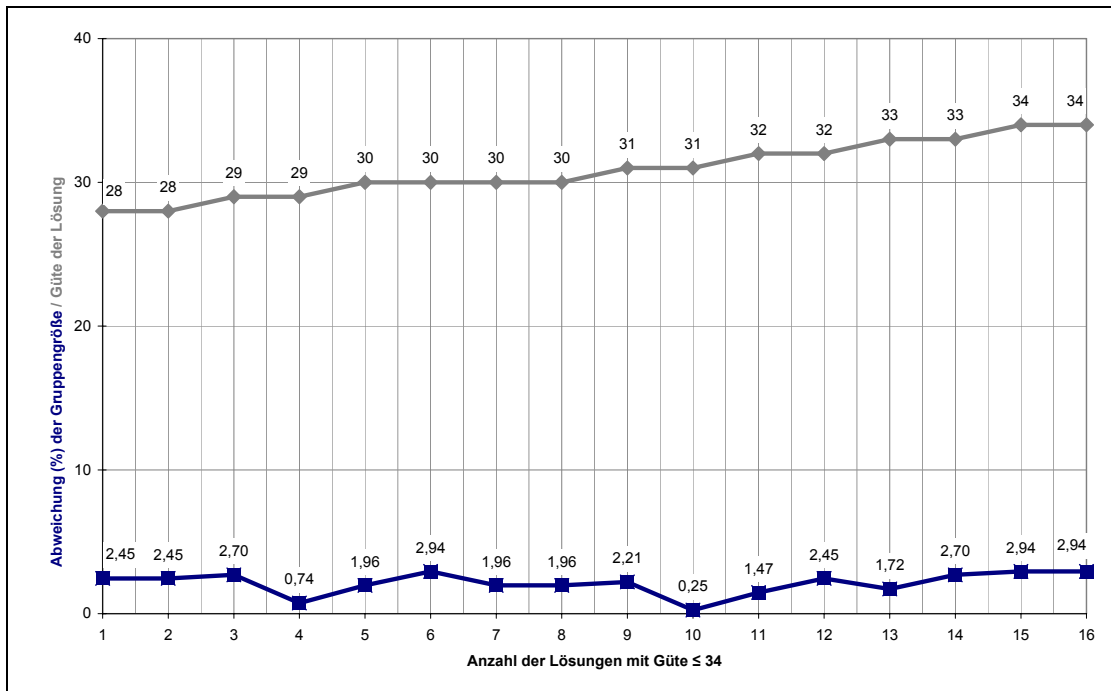


Abb. 7.58 Performance der 16 besten Lösungen





Abb. 7.59 Graphische Darstellung der Lösungen 1, 2, 4 und 10

Unter Verwendung von Straftermen wird die Messung der Gesamtpformance wie folgt vorgenommen:

$$\text{Gesamtpformance} = S_g * \text{Güte} + S_a * \text{Abw.} * \text{Gesamtgruppengröße}$$

$S_g$  = Strafterm zur Gewichtung der Güte

$S_a$  = Strafterm zur Gewichtung der Gesamtgruppengröße

Güte = Absolute Abweichung ( $\text{m}^2$ ) jeder Fläche von der im Raumprogramm geforderten Fläche

Abw. = Prozentuale Abweichung von der Gesamtgruppengröße

Nach Maßgabe des Architekten bestimmen die Strafterme die Gewichtung der Güte und der Gesamtgruppengröße. Wird ein Wert mit einem positiven Strafterm größer 1 belegt, kommt diesem bei der Ermittlung der Gesamtpformance ein höheres Gewicht zu. Tab. 7.26 und Abb. 7.60 zeigen die Auswertung der Ergebnisse ohne Gewichtung. Tab. 7.27 und Abb. 7.61 verdeutlichen die Veränderungen durch ein höheres Gewicht für den Wert der Güte. In beiden Fällen wird deutlich, dass die Lösungen 4 und 10 die geringsten Werte aufweisen. Bei gewichteter Verteilung, d.h. in diesem Fall einer höheren Gewichtung der Güte, erzielt Lösung 4 den geringsten Wert und demzufolge die beste Gesamtpformance (Abb. 7.62).



	Summe	Güte (m²)	Ann. (%)	Abw. (%)	Performance
Fläche Lösung 1	398	28	97,55	2,45	38
Fläche Lösung 2	398	28	97,55	2,45	38
Fläche Lösung 3	397	29	97,30	2,70	40
Fläche Lösung 4	405	29	99,26	0,74	32
Fläche Lösung 5	400	30	98,04	1,96	38
Fläche Lösung 6	396	30	97,06	2,94	42
Fläche Lösung 7	400	30	98,04	1,96	38
Fläche Lösung 8	400	30	98,04	1,96	38
Fläche Lösung 9	399	31	97,79	2,21	40
Fläche Lösung 10	407	31	99,75	0,25	32
Fläche Lösung 11	402	32	98,53	1,47	38
Fläche Lösung 12	398	32	97,55	2,45	42
Fläche Lösung 13	401	33	98,28	1,72	40
Fläche Lösung 14	397	33	97,30	2,70	44
Fläche Lösung 15	396	34	97,06	2,94	46
Fläche Lösung 16	396	34	97,06	2,94	46
sizeRooms	408	0	100,00	0,00	0
Strafterm S_g	1				
Strafterm S_a	1				

Tab. 7.26 Gesamtperformance ohne zusätzliche Straftermgewichtung

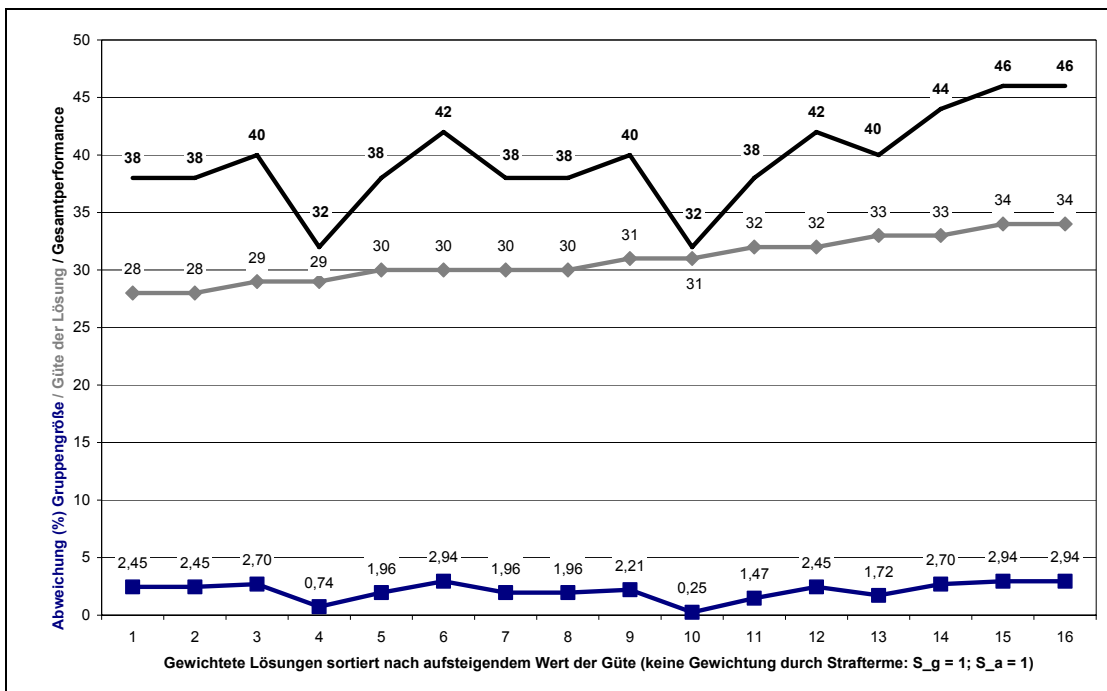


Abb. 7.60 Gesamtperformance ohne zusätzliche Straftermgewichtung

	Summe	Güte (m²)	Ann. (%)	Abw. (%)	Performance
Fläche Lösung 1	398	28	97,55	2,45	66
Fläche Lösung 2	398	28	97,55	2,45	66
Fläche Lösung 3	397	29	97,30	2,70	69
Fläche Lösung 4	405	29	99,26	0,74	61
Fläche Lösung 5	400	30	98,04	1,96	68
Fläche Lösung 6	396	30	97,06	2,94	72
Fläche Lösung 7	400	30	98,04	1,96	68
Fläche Lösung 8	400	30	98,04	1,96	68
Fläche Lösung 9	399	31	97,79	2,21	71
Fläche Lösung 10	407	31	99,75	0,25	63
Fläche Lösung 11	402	32	98,53	1,47	70
Fläche Lösung 12	398	32	97,55	2,45	74
Fläche Lösung 13	401	33	98,28	1,72	73
Fläche Lösung 14	397	33	97,30	2,70	77
Fläche Lösung 15	396	34	97,06	2,94	80
Fläche Lösung 16	396	34	97,06	2,94	80
sizeRooms	408	0	100,00	0,00	0
Strafterm S_g	2				
Strafterm S_a	1				

Tab. 7.27 Gesamtperformance mit zusätzlicher Straftermgewichtung

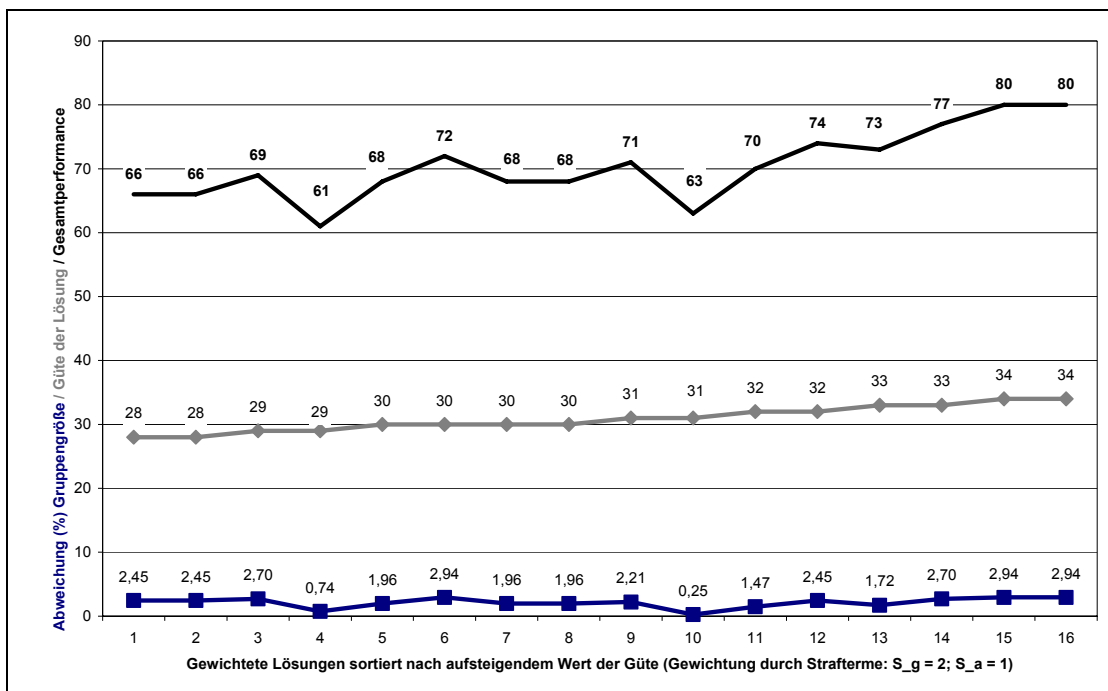


Abb. 7.61 Gesamtperformance mit zusätzlicher Straftermgewichtung



Abb. 7.62 Soll/Ist-Vergleich in m². Lösung 4 - Beste Gesamtperformance

### 7.3.5 Bewertung

Die Lösung komplexer kombinatorischer Optimierungsprobleme ist mit Sprachen wie OPL möglich. Da die in der Optimierung von Flächenanordnungen existenten Probleme jedoch in der Regel NP-schwer sind, ist eine erfolgreiche Problemlösung, d.h. die Vermeidung der kombinatorischen Explosion, die Annäherung an ein Optimum und die Erzeugung einer plausiblen Lösung, nur durch die Verwendung geeigneter Suchstrategien zu erreichen. Das die Definition der Suchstrategie in Abhängigkeit von der Problemstellung variiert, stellt dabei ein schwerwichtiges Problem für die erfolgreiche Implementierung von Optimierungsverfahren in Architekturprogramme dar. Es ist nur schwer vorstellbar, dass ein Architekt ohne fundierte Kenntnisse der Methoden des Operations Research eine Suchstrategie entwickeln kann, welche die Lösung einer Optimierungsaufgabe unter geringen Kosten gewährleistet. Die Stärke von OPL begründet sich jedoch insbesondere dadurch, dass es möglich ist, Einfluss auf die Suchrichtung eines Optimierungsproblems zu nehmen, ohne den architektonischen Lösungsraum dabei einschränken zu müssen. Nur die gezielte Definition von Search Procedures ermöglicht eine effektive Lösungssuche, die für die praxismgerechte Implementierung innerhalb der Architektur notwendig ist. Die auf den entwickelten mathematischen Modellen basierenden Deklarationen der Variablen, Konstanten und Nebenbedingungen können problemlos in einer Programmarchitektur abgebildet werden, da ihr Design klar definierten Regeln folgt. Die Abbildung von Search Procedures ist sehr viel schwieriger zu realisieren, weil deren Struktur in Abhängigkeit vom Optimierungsproblem stark variiert. Bei den Search Procedures handelt es sich eben nicht um Default-Strategien zur Lösungssuche. Diese Default-Strategien sind selbstverständlich auch in den Solvern von OPL integriert und sind bei weniger komplexen, oder "effizient lösbaren" Problemen sehr leistungsstark. NP-schwere Probleme, also "schwierige" oder "schwer lösbare" Probleme, jedoch können - wenn überhaupt - nur mit einer auf die jeweilige Problemdefinition zugeschnittenen Strategie in angemessener Zeit gelöst werden.

Es ist möglich, bei architektonischen Problemen gewisse Hilfestellungen für die Vorgehensweise bei der Definition dieser Search Procedures für das ZR-Modell zu geben. Eine allgemeingültige Strategie kann es, auf Grund der Struktur dieser Probleme, jedoch nicht geben. Die Hilfestellungen beziehen sich bspw. auf die proportionale Ausprägung der Units. Die Definition der Proportionen kann bereits als Nebenbedingung definiert werden, eine Deklaration im Search Teil des Optimierungsproblems ist aber zweifelsfrei effektiver, da der Solver zunächst den Lösungsraum mit Hilfe dieser Deklarationen durchsucht. Wird keine optimale Lösung gefunden, wird der Constraint Store des Problems gelöscht, und der Solver verwendet die OPL Default-Strategien zur Fortsetzung der Lösungssuche. Der Einsatz geeigneter Search-Procedures verringert die Berechnungskosten eines Optimierungsproblems in entscheidendem Maße, er birgt jedoch auch die Gefahr, dass bei fehlerhafter Deklaration mögliche Lösungen eines Problems nicht gefunden werden. Es ist fest zu halten, dass es beim korrekten Einsatz der Search Procedures nur um die mit der Berechnung verbundenen Kosten geht. Der Solver findet, sofern es auf der Grundlage der mathematischen Modelle eine Lösung für das Optimierungsproblem gibt, in jedem Fall eine Lösung. Die Kosten einer Berechnung können ebenfalls durch die Einschränkung des Flächenverbrauchs einer Unit verringert werden. Die Deklaration der Flächengröße in

Quadratmetern ist ein Beispiel für eine derartige Einschränkung. Sie ist auch architektonisch sinnvoll, da es sich um eine Angabe handelt, welche in Raumprogrammen zu finden ist. Weitaus effektiver ist aber die variable Deklaration der Länge und Breite einer Unit mit Hilfe Oberer und Unterer Schranken. Am effektivsten ist unzweifelhaft die Definition einer Unit durch die konstante Deklaration der Seitenlängen. Diese Vorgehensweise widerspricht jedoch dem eigentlichen Ziel der Verwendung von Optimierungsverfahren bei der Neuordnung von Flächen in der Revitalisierung. Hier muss größtmögliche Variabilität bei der Anordnung und geometrischen Ausprägung der Flächen gewährleistet werden, damit innerhalb einer gegebenen Grundfläche eine Lösung für ein Problem gefunden werden kann, dessen Prämisse der geringstmögliche Eingriff in die bauliche Ausprägung des Bestandes ist.

Die Evaluation des OPL-Prototyps zeigt, dass zusätzlich zur Wahl der geeigneten Programmiersprache auch die Art der programmiertechnischen Umsetzung ein gewichtiges Kriterium für die erfolgreiche Verwendung des ZR- oder ZA-Modells ist. In OPL wurden für beide Modelle unterschiedliche programmiertechnische Vorgehensweisen auf der Basis der entwickelten mathematischen Modelle evaluiert. Es zeigte sich, dass unterschiedliche Programmstrukturen für ein und dasselbe mathematische Modell zu völlig unterschiedlichen Berechnungskosten führen können. Es wurde ebenfalls deutlich, dass die Entwicklung eines mathematischen Modells zur Darstellung eines Optimierungsproblems im Idealfall zwar unabhängig von der Verwendung einer bestimmten Programmiersprache ist, die tatsächliche Leistungsfähigkeit einer Sprache allerdings erst durch eine enge Abstimmung von mathematischem und programmiertechnischem Modell zum Tragen kommt. So spielte bspw. bei der Evaluation des ZR-Modells und der Verwendung des mathematischen Modells (Kap. 6.1.2) die Reihenfolge der Variablendeklaration eine entscheidende Rolle. Die Evaluation eines Prototyps ohne Search Procedures mit identischen Constraints und Zielfunktionen unter Verwendung unterschiedlicher Reihenfolgen der Variablendeklaration zeigte, dass der Solver beginnt, mit den zuerst deklarierten Variablen eine Teillösung des Constraintstapels herbeizuführen. Handelt es sich um Variablen mit starker Abhängigkeit von anderen Variablen, erhöhen sich die Berechnungskosten dadurch erheblich. Theoretisch bedingt diese Eigenschaft, dass die Variablendeklaration wohlbedacht und in einer bestimmten Reihenfolge in Form eines Eröffnungsverfahrens, welches eine näherungsweise geometrische Anordnung der Units vorgibt, durchgeführt werden muss. Diese Vorgehensweise steht jedoch im Widerspruch zum gewünschten Vorgehen beim Einsatz eines Optimierungsverfahrens an sich. Das Problem kann durch die Ergänzung der OPL Default Suchstrategie mit Hilfe von Search Procedures gelöst werden.

Zusammenfassend lässt sich festhalten, dass mit OPL Aufgaben die aus der Anwendung des ZR- und ZA-Modells resultieren, erfolgreich gelöst werden können. Dazu stehen sowohl Mechanismen der Constraint Programmierung als auch der Integer- und Nicht-Linearen Programmierung zur Verfügung. Abhängig von der Wahl des Solvers, können eine (CPLEX MIP, Solver MIP) oder mehrere (Solver, Solver + Hybrid, Solver + Hybrid + Scheduler) Lösungen für ein Optimierungsproblem gefunden werden. Durch die Verwendung von Search Procedures kann aktiv in die Vorgehensweise der Optimierungsalgorithmen eingegriffen werden. Diese Eingriffe können die Berechnungskosten eines Problems dramatisch reduzieren. Sie können jedoch unter Umständen auch dazu führen, dass potentielle Lösungen eines Optimierungsproblems nicht gefunden werden. Neben einer effektiven mathematischen Modellierung und der entsprechenden programmiertechnischen Umsetzung, stellt die Definition dieser Search Procedures ein schwer-

wiegendes Problem bei der Verwendung von OPL zur Lösung von Umbau- oder Umnutzungsaufgaben durch Architekten dar (vgl. Kap. 8).

Vielfach wird die Schwere eines zu lösenden Problems und die Leistungsfähigkeit eines Solvers in Abhängigkeit von der Anzahl der berechenbaren Variablen und Constraints definiert. Diese Abhängigkeiten finden sich in der Regel auch in den Prototypen wieder. Allerdings muss angeführt werden, dass die Komplexität eines Problems nicht ausschließlich auf diesen Parametern beruht. Problemstellungen für ZR-Modelle, weisen in den Prototypen teilweise weniger Variablen und Constraints auf, können jedoch mit höheren Berechnungskosten verbunden sein. Dieses Verhalten ist damit zu begründen, dass in den ZA-Modellen die ermittelten Teilgraphen in einer aufwändigen Berechnung auf Zusammenhang geprüft werden müssen. Es sollte auch angeführt werden, dass Berechnungen durchgeführt wurden, von welchen die Existenz einer perfekten Lösung bekannt war, diese jedoch von den Algorithmen auch nach mehreren Tagen Berechnungszeit nicht gefunden werden konnte. Die Ursache dafür ist unklar. Da die Funktionsfähigkeit des mathematischen und programmiertechnischen Modells, auf Grund der erfolgreichen Evaluation derselben Problemstellung mit unterschiedlichen Variablenwerten, uneingeschränkt gegeben ist, kann die Ursache dafür nicht spezifiziert werden.

## **8 Zusammenfassung und Ausblick**

### **8.1 Zusammenfassung**

Die Bauaufgaben der Zukunft liegen in der Auseinandersetzung mit bestehender Architektur. Der Verzicht auf den Neubau zu Gunsten des Umbaus oder der Umnutzung ist aus ökonomischen, ökologischen und sozialen Gründen geboten. Vorhandene Untersuchungen zur Kategorisierung des Bestandes zeigen heterogene Gefüge, aus welchen sich jedoch zwei Gebäudetypen für die vorliegende Arbeit extrahieren ließen. Diese sind Gebäude mit unflexiblen Grundrissorganisationen und massiven Konstruktionen, sowie Gebäude mit flexiblen Grundrissorganisationen und leichten Konstruktionen. Diese Gebäudetypen stellen auf Grund ihrer Größe eine besondere Herausforderung an den Umbau oder die Umnutzung dar. Sie bergen ein Revitalisierungspotential, welches insbesondere in den neuen Bundesländern zur Wiederbelebung innerstädtischer Brachflächen genutzt werden muss.

Da Gebäude mit leichten Konstruktionen unter geringen Eingriffen in die Bausubstanz verändert werden können, unterliegen diese Gebäude schnell baulichen Veränderungen. Doch auch Gebäude mit massiven Konstruktionen werden unter erheblichen Eingriffen in die Bausubstanz neuen Nutzungen zugeführt. Beide Vorgehensweisen sind, wenn es nicht zur vollständigen Entkleidung des Bestandes kommt, eine bessere Alternative als der Neubau. Doch nicht ausschließlich Merkmale der Konstruktion eines Gebäudes bestimmen dessen Chancen einer Revitalisierung zugeführt zu werden. Oftmals sind es auch Unsicherheitsfaktoren oder Eitelkeiten auf planerischer Seite, die nur allzu schnell die Wahl zu Gunsten des Neubaus ausfallen lassen. Dieser Umstand ist offenkundig auch der Tatsache geschuldet, dass es kaum planerische Werkzeuge gibt, die den Architekten während der frühen Phase seiner Planungstätigkeit bei der Entscheidungsfindung Für und Wider einer Revitalisierung unterstützen. Zu viele Faktoren determinieren diesen Prozess, als dass es dem Architekten möglich wäre, bereits zu Beginn der Planungs-

tätigkeit eine plausible Aussage zu treffen ob ein Gebäude revitalisierbar ist oder nicht. Ferner werden alternative Planungsmethoden, die nicht notwendigerweise das Bauen in den Vordergrund stellen, nur selten angewendet. Die Tätigkeit des Architekten fokussiert eher den entwerferischen als den planerischen Aspekt, so dass Revitalisierungsstrategien wie der Flächentausch im Grunde genommen nicht zum Repertoire seiner Planungstätigkeit gehören. Da unter der Prämisse des nachhaltigen Umgangs mit dem Gebäudebestand die Weiternutzung eines Gebäudes unter größtmöglicher Beibehaltung der existenten Bausubstanz und unter geringst möglicher Veränderung des baulichen Zustands erreicht werden muss, besitzt jedoch gerade diese Strategie außerordentliches Potential. Ihr enormer Nutzwert besteht in der Übertragbarkeit auf Flächen jeglicher Größenordnung. Voraussetzung dafür ist die Existenz geeigneter Datenbanken, die potentielle Revitalisierungsobjekte mit ihren spezifischen Merkmalen und dem Angebot der tauschbaren Flächen beinhalten.

Letztlich bestimmt der praktizierende Architekt in entscheidendem Maße, ob ein Gebäude revitalisiert wird oder nicht. Zur Unterstützung dieser Entscheidung wurde in der vorliegenden Arbeit die Annahme getroffen, dass ein wesentliches Kriterium für die Entscheidungsfindung die Erfüllung der Anforderungen des Raumprogramms ist. Dafür wird die Anpassbarkeit der existenten Grundrisskonstellation des Bestandsgebäudes an die Erfordernisse des Raumprogramms geprüft. Diese Überprüfung, die Gegenstand der vorliegenden Arbeit ist, stellt im Wesentlichen eine kombinatorische Problemstellung dar, die äußerst komplex ist. Es wurde aufgezeigt, dass bereits die Neuordnung weniger Flächen eine extrem hohe Zahl möglicher Kombinationsmöglichkeiten aufweist. Zur Unterstützung dieser Vorgehensweise wurden zwei Modelle entwickelt, welche sich an den Eigenarten der zuvor genannten massiven und leichten Konstruktionen orientieren. Umbauten werden dabei durch die Zerstörungsreiche Modellbildung (ZR) repräsentiert, während Umnutzungen mit Hilfe der Zerstörungsarmen Modellbildung (ZA), insbesondere den Flächentausch, dargestellt werden. Analogien zu den Anforderungen die an diese Modelle gestellt werden, finden sich in vielen Bereichen des täglichen Lebens. In Warenlagern, Containerterminals, in Brettspielen oder auch in der Terminplanung. Den Analogien ist gemein, dass es informationstechnische Methoden gibt, mit deren Hilfe diese repräsentiert und deren inhärente Anforderungen gelöst werden können. Dazu eignen sich Methoden des Operations Research, welche die realen Entscheidungsprobleme in Optimierungsprobleme überführen und mit Hilfe leistungsstarker Algorithmen lösen. Die in der vorliegenden Arbeit erläuterte Modellbildung, die daraus resultierenden Optimierungsmodelle, sowie die aufgestellten mathematischen Modelle und die entwickelten Prototypen orientieren sich an dieser Vorgehensweise.

Die Berechnungen der Prototypen von Aufgaben der Zerstörungsreichen und Zerstörungsarmen Modelle belegen, dass mit ihnen Revitalisierungsprobleme der Grundrissplanung gelöst werden können. Normative Einflüsse und Entwurfsziele können in Form von Nebenbedingungen und Zielfunktionen definiert werden, die in die Optimierungsrechnung einfließen. Die Ergebnisse sind insbesondere hinsichtlich der Erzielbarkeit einer großen Anzahl unterschiedlicher Lösungen, die allesamt die gleichen Anforderungen erfüllen, viel versprechend. Diese große Anzahl ermittelbarer Lösungen belegt den potentiellen Einsatz des Werkzeugs: die Berechnung einer großen Anzahl perfekter Lösungen eines Problems, die automatisierte Messung der Performance bei den ZA-Modellen und letztlich den manuellen Abgleich der besten Lösungen durch den Architekten, welcher auf der Basis dieser Lösungen seine Planungstätigkeit fortsetzt.



Aus den objektiv bewerteten Lösungen kann ein Architekt relativ schnell jene Lösungen extrahieren, die eine für ihn besondere Qualität aufweisen. Dabei ist der Begriff der Qualität durchaus so zu verstehen, dass diese Lösungen qualitativ nicht objektiv messbar sein müssen. Alle ermittelten Lösungen erfüllen, auf Grund der Berechnung der Nebenbedingungen und Zielfunktionen, sowie der Performance-Messung, die an sie gestellten Anforderungen. Die daraus letztlich gewählte und zur Ausführung gelangende Lösung kann subjektiven Kriterien unterliegen.

## 8.2 Bewertung

Folgende Ergebnisse können aus der vorliegenden Arbeit abgeleitet werden:

Die Prototypen CATIA und VisualDOC sind für die Revitalisierung untauglich. Auf der Grundlage der entwickelten geometrischen und mathematischen Modelle ist es mit ihnen lediglich möglich, Optimierungsprobleme extrem geringer Komplexität zu lösen. Derartige Probleme können vom Architekten mühelos ohne den Einsatz von Optimierungsverfahren bearbeitet werden. Während für den Prototyp CATIA überwiegend das Problem der Lokalen Optima unlösbar ist, stellt für den Prototyp VisualDOC im Wesentlichen die Kombination mehrerer Nebenbedingungen das schwerwiegendste Problem dar. Der Prototyp OPL ist der einzige Prototyp, mit welchem komplexe Probleme bearbeitet und gelöst werden konnten. Ferner war es nur mit diesem Prototyp möglich, die Umsetzung des Zerstörungssarmen Modells, sowie die in dieses Modell integrierte Performance-Messung, zu realisieren. Die nachfolgenden Ergebnisse beziehen sich daher ausschließlich auf diesen Prototyp.

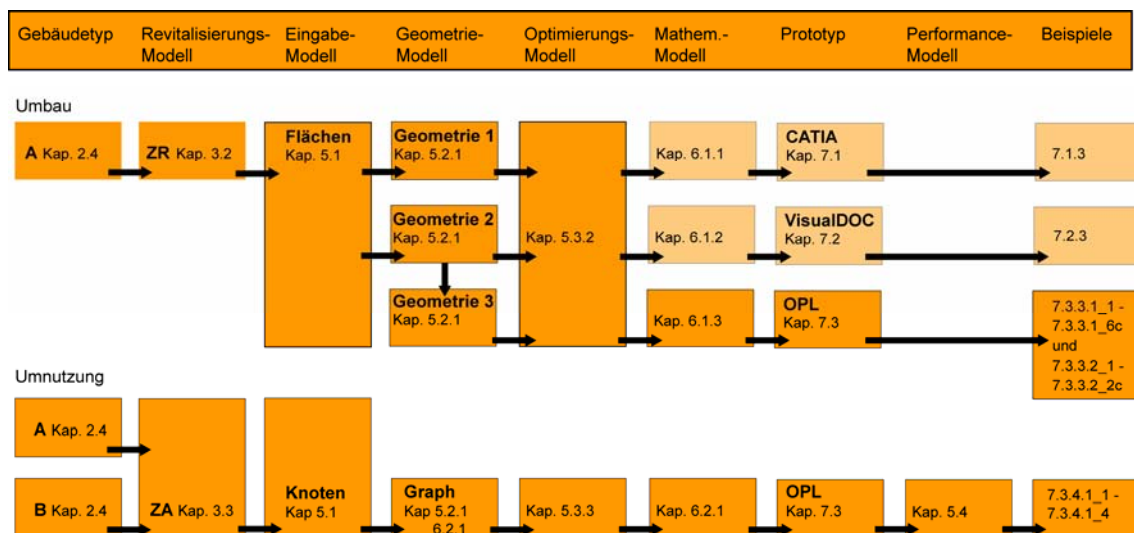


Abb. 8.1 Zusammenfassende Übersicht der entwickelten Modelle und Prototypen

### **Zerstörungsreiches Modell:**

- Das ZR-Modell kann mit Hilfe der entwickelten geometrischen Modelle repräsentiert werden.
- Es wurden zwei geometrische Repräsentationen (Kap. 6.1.1 und 6.1.2) entwickelt, die in der Definition der Referenzpunkte, der Länge und Breite der Units, sowie der damit verbundenen Response-Variablen differieren. Keine der beiden Repräsentationen besitzt wesentliche Merkmale, die sich besonders vorteilhaft oder nachteilig auf die Optimierungsrechnung auswirken.
- Aufbauend auf der zweiten geometrischen Repräsentation (Kap. 6.1.2) wurde eine dritte geometrische Repräsentation (Kap. 6.1.3) entwickelt, die es erlaubt, die Grundstücksgrenzen in die Optimierungsrechnung mit einzubeziehen. Durch diese Erweiterung wird der Aufbau von Räumen an den bestehenden Gebäudegrundriss ermöglicht.
- Es wurden drei mathematisch differente Modelle (Kap. 6.1.1 - 6.1.3) für die geometrische Repräsentation entwickelt. Die im Ausgangsmodell (Kap. 6.1.1) verwendeten MIN-Funktionen erwiesen sich in der Optimierungsrechnung als nachteilig. Sie wurden in den nachfolgenden Modellen durch Boolesche Variablen ersetzt. Diese Modelle (Kap. 6.1.2 - 6.1.3) zeigten in den Optimierungsrechnungen größere Effizienz.
- Wichtiger als die geometrische und mathematische Repräsentation ist die programmiertechnische Umsetzung des Optimierungsproblems. Es wurden in Beispielrechnungen unterschiedliche programmiertechnische Verfahren eingesetzt (Lineare-, Nicht-Lineare-, Integer-, Mixed-Integer-, Constraint Programmierung), die im Wesentlichen unabhängig von der architektonischen Problemdefinition waren. Die Berechnungskosten und Ergebnisse differieren massiv. Die richtige Problemdefinition und die entsprechende Wahl des programmiertechnischen Verfahrens sind daher von besonderer Bedeutung für effektive Berechnungen mit geringen Kosten. Zur Reduzierung dieser Kosten wurden mit Hilfe von Approximationen nicht-lineare Probleme in lineare überführt. Probleme der Integer-Programmierung können auf Grund des Ganzzahligkeitserfordernisses nur mit Hilfe von Search Procedures effektiv gelöst werden.
- Mit dem Prototyp konnten komplexe architektonische Probleme der ZR-Modelle gelöst werden. Unter „komplex“ sei hier eine Problemstellung verstanden, die ein Architekt nur mit erheblichem Aufwand lösen könnte, ohne jedoch sicher zu sein, den gesamten potentiellen Lösungsraum durchsucht und tatsächlich eine perfekte Lösung erzielt zu haben.
- Mit dem Prototyp können alle perfekten Lösungen eines Optimierungsproblems erzeugt werden, die zweifelsfrei von einem Architekten nicht erarbeitet werden können.

### **Zerstörungsarmes Modell:**

- Das ZA-Modell kann graphentheoretisch repräsentiert werden.
- Es wurde eine graphentheoretische und eine dazugehörige mathematische Repräsentation entwickelt (Kap. 6.2.1).
- Die graphentheoretische Abbildung des Modells ermöglicht die Repräsentation der Units in Form von Knoten innerhalb eines Graphen. Mit Hilfe der Graphen ist es möglich, 2-dimensionale und 3-dimensionale räumliche Strukturen abzubilden. Eine 2-dimensionale

- Matrix kann auf eines oder mehrere Geschosse, die sich auch in unterschiedlichen Gebäuden befinden können, angewendet werden.
- Die Verwendung von Blockmatrizen ermöglicht die Abbildung ganzer Gebäude oder Liegenschaften, einschließlich der darin enthaltenen Geschosse.
  - Mit dem Prototyp konnten extrem komplexe architektonische Probleme der ZA-Modelle gelöst werden. Unter „extrem komplex“ sei hier eine Problemstellung verstanden, die ein Architekt in annehmbarer Zeit nicht lösen könnte.
  - Mit dem Prototyp können alle perfekten Lösungen eines Optimierungsproblems erzeugt werden, die zweifelsfrei von einem Architekten nicht erarbeitet werden können.
  - Der Prototyp integriert die Verwendung gewichteter Zielfunktionen, die eine nicht vollständige Erfüllung der Zielfunktionen berücksichtigen.
  - Die Anzahl der ermittelbaren Lösungen ist abhängig von der Granularität der Netztopologie. Je feiner das Netz, d.h. je höher die Anzahl der die Knoten verbindenden Kanten, desto größer ist die Wahrscheinlichkeit, eine perfekt Lösung zu erzielen. Die Performance der ermittelbaren Lösungen ist daher in entscheidendem Maße davon abhängig, wie groß die Bereitschaft des Architekten ist, das Netz zu verfeinern. Diese Verfeinerung des Netzes kann zu größeren Abständen zwischen einzelnen Raumeinheiten einer ermittelten Lösung führen, ist jedoch unter der aufgestellten Prämisse des ressourcenschonenden Umgangs mit dem Bestand empfehlenswert.
  - Die Performance der ermittelten Lösungen wird automatisch gemessen.

### **Für beide Modelle gilt:**

- Die Anzahl der Variablen und Constraints eines Optimierungsproblems ist keine absolute Kenngröße für die zu erwartenden Berechnungskosten. Die berechneten Beispiele zeigen, dass diverse komplexe Probleme mit Hunderten von Variablen in weniger als einer Sekunde gelöst werden konnten, wohingegen Probleme mit wenigen Dutzend Variablen, Stunden zur Lösung benötigten.
- Die Berechnungskosten sind ebenfalls keine absolute Kenngröße für die Leistungsfähigkeit eines Solvers. Hier gilt, dass ein Problem mit geringeren Kosten gelöst werden kann, wenn eine explizit auf die Problemstellung zugeschnittene programmiertechnische Umsetzung gegeben ist.
- Die Berechnungskosten eines Problems lassen sich durch eine effektive Modellierung und die damit verbundene Wahl des richtigen programmiertechnischen Verfahrens reduzieren. Eine effektive Modellierung beinhaltet eine elegante Deklaration der Problemstellung, sowie die Verwendung von Search Procedures.
- Search Procedures sind die einzigen Konstrukte, welche die effektive Lösung eines komplexen kombinatorischen Optimierungsproblems gewährleisten. Ihre Verwendung ist bei NP-schweren Problemen zwingend notwendig. Die korrekte Anwendung hat immer eine drastische Reduzierung der Berechnungskosten zur Folge. Die falsche Deklaration einer Search Procedure kann dazu führen, dass der zu durchsuchende Lösungsraum derart reduziert wird (Domain Reduction), dass potentielle Lösungen nicht gefunden werden können.

- Die Verwendung einer Search Procedure ist dann effektiv, wenn sie auf jene Variablen angewendet wird, zu welchen eine große Anzahl anderer Variablen Abhängigkeiten aufweisen.
- Gebäude mit flexiblen Grundrissorganisationen und leichten Konstruktionen, die durch Systemlinien oder Raster unterteilt werden können und eine hohe Komplexität aufweisen, eignen sich grundsätzlich zur Anwendung des Zerstörungsreichen und des Zerstörungsarmen Modells. In jedem Fall sollte zunächst das Zerstörungsarme Modell Anwendung finden. Auch eine Kombination beider Modelle ist denkbar. Das Zerstörungsreiche Modell sollte nur dann zur Anwendung gelangen, wenn große Freiräume beplant werden müssen, die nicht augenscheinlich durch Systemlinien oder Raster unterteilt werden können.

### **Nicht verifizierte Aussagen und offene Problemstellungen:**

In der vorliegenden Arbeit wurde ein Systemkonzept vorgestellt, welches durch den Einsatz von Optimierungsverfahren in der Grundrissplanung Lösungen für Umbau- und Umnutzungsaufgaben automatisiert erzeugt. Es wurden unterschiedliche Lösungsansätze aufgezeigt und deren prototypische Umsetzung demonstriert. Eine praxiserichte Umsetzung ist möglich, wenn die einzelnen Lösungsansätze von Fachkräften der jeweiligen Fachdisziplinen evaluiert und aufgearbeitet werden.

Die Erprobung des OPL-Prototyps zeigt, dass komplexe architektonische Probleme mit den erarbeiteten Modellen lösbar sind. Offen ist die Frage, welche Komplexität diese Probleme annehmen dürfen, so dass sie mit vertretbaren Berechnungskosten lösbar sind. Damit verbunden sind die Fragen, wie die Komplexität derartiger Optimierungsprobleme allgemeingültig gemessen werden kann, und wie jene Parameter gezielt aus den Optimierungsproblemen extrahiert werden können, die wesentlichen Einfluss auf den Berechnungsverlauf nehmen.

Es steht außer Frage, dass der Vergleich des Raumprogramms mit dem Gebäudegrundriss nur einer von vielen Faktoren bei der Entscheidungsfindung für oder gegen eine Revitalisierungsmaßnahme ist. Es ist daher zu untersuchen, welche weiteren Faktoren als Parameter in die Optimierungsrechnung einfließen müssen, um das Revitalisierungspotential eines Gebäudes gezielter und plausibler darstellen zu können.

### 8.3 Ausblick

Der nachfolgende Ausblick ist als Szenario für die Erweiterung der erarbeiteten Systemkonzeption zu verstehen. Die beschriebenen Kriterien werden auf der Grundlage der erzielten Ergebnisse als bedeutend für die praxisgerechte Anwendung von Optimierungsverfahren zur Lösung von Umbau- und Umnutzungsaufgaben angesehen. Aus ihnen können Anregungen für weitere Forschungsarbeiten entnommen werden.

#### **Eingabemodell (Zerstörungsreiches Modell)**

Das wichtigste Kriterium für die Praxistauglichkeit der Systemkonzeption ist die Integration einer Komponente zur automatisierten Erzeugung von Datensätzen für das Eingabemodell. Die in den Prototypen verwendeten Daten der Raumprogramme entsprechen einer konkreten Planungsanfrage, d.h. sie repräsentieren eine Situation, bei der überprüft wird, ob ein existentes Raumprogramm in eine konkrete baulich veränderbare Umgebung integriert werden kann. Die in den Raumprogrammen enthaltenen Daten sind jedoch unvollständig. Insbesondere die in den Optimierungsmodellen Anwendung findenden Nebenbedingungen in Form von objektiven und subjektiven Regeln muss der Architekt, aufbauend auf Erinnerung und Erfahrung oder dem Vergleich mit anderen Bauvorhaben, eigenständig integrieren. Diese Vorgehensweise birgt die Gefahr, dass etablierte Regeln aus möglicher Unkenntnis missachtet und somit potentiell richtige Lösungen der Optimierungsaufgabe ausgespart bleiben. Es ist jedoch gewiss, dass etablierte Regeln in vergleichbaren Bauaufgaben bereits vielfach angewendet wurden. Es wird daher die Extraktion allgemeingültiger Regeln aus Beispielen verschiedener Nutzungstypologien angestrebt. Diese Regelextraktion kann möglicherweise durch den Einsatz Künstlicher Neuronaler Netze erfolgen. Es existieren Netztopologien, welche durch Funktionsapproximation Zusammenhänge zwischen den Knoten der Netze erkennen und erfolgreich auswerten können [Rieger, 2002]. Ziel ist die Erzeugung charakteristischer Stammdatensätze zu den Eigenschaften und Beziehungen der Räume unterschiedlicher Nutzungstypologien.

#### **Eingabemodell (Zerstörungsarmes Modell)**

Eine ähnliche Problemstellung liegt bei der Verwendung des Zerstörungsarmen Modells vor. Hier wird ebenfalls überprüft, ob eine konkrete Zielvorgabe in Form eines Raumprogramms in eine existente Struktur eingepasst werden kann. Auch auf diese Aufgabe kann die vorgenannte Vorgehensweise zur Erzeugung von Stammdatensätzen zu den erforderlichen Eigenschaften und Beziehungen zwischen Räumen angewendet werden. Der Abgleich eines konkreten Raumprogramms mit einer existenten, nicht veränderbaren Raumstruktur führt jedoch nur äußerst selten zu einer vollständigen Übereinstimmung. Aus diesem Grund wurden in den entwickelten Prototypen gewichtete Zielfunktionen zur Bestimmung der Güte einer Planungslösung eingeführt. Eine rekursive Vorgehensstrategie kann hier zu einer potentiellen Erweiterung des Lösungsraums führen. Bei dieser Strategie werden die existenten räumlichen Strukturen der Revitalisierungsmaßnahme mit den Stammdatensätzen unterschiedlicher Nutzungstypologien abgeglichen. Diese Vorgehensweise findet bspw. in Künstlichen Neuronalen Netzen zur Mustererkennung

(Pattern Recognition) Anwendung [Lämmel und Cleve, 2001]. Das Ziel dieser Maßnahme ist, die Messung der räumlichen Übereinstimmung der Ursprungsnutzung mit den räumlichen Anforderungen einer potentiellen Nachnutzung zu erreichen (Pattern Matching). Anders als bei der üblichen Vorgehensweise wird hier also nicht untersucht, wie gut „welche konkrete“ Nachnutzung in die existente Struktur eingebunden werden kann, sondern welche konkrete Nachnutzung sich am besten zur Einbindung in die existente Struktur eignet. Diese Vorgehensweise ist ebenfalls auf das Raumprogramm des Zerstörungsreichen Modells anwendbar.

### **Geometriemodell**

Die Verknüpfung der Optimierungsrechnung mit einer geometrischen Abstraktion ist für die interaktive Benutzerführung unabdingbar. Eine Implementierung von Optimierungsumgebungen muss daher in die dem Architekten vertraute Planungsumgebung erfolgen. Das parametrisch-assoziative Paradigma aktueller CAAD-Systeme und die Existenz geeigneter Schnittstellen bildet dafür eine ausreichende Grundlage. Dieses Paradigma erlaubt auch die Verwendung der entwickelten Geometriemodelle ohne eine Anpassung an die geometrische Repräsentation der Zielsysteme vornehmen zu müssen. Solver können durch die Nutzung der API's problemlos in CAAD-Programme integriert werden. Auch die Einbettung in Webbasierte Applikationen ist möglich. Das Geometriemodell muss ferner die automatisierte Erfassung von Daten für das Eingabemodell unterstützen. Hier ist insbesondere die Erzeugung der Matrizen des Zerstörungsarmen Modells zu nennen, welche auf manuellen Weg fehleranfällig und extrem zeitaufwändig ist. Eine Realisierung dieser Funktionalität kann mit Hilfe der IFC umgesetzt werden.

### **Optimierungsmodell**

Die entwickelten Optimierungsmodelle sind erweiterbar. Objektive und Subjektive Regeln können jederzeit in die Modelle integriert werden. Besonderes Potential kommt dabei den zur Anwendung gelangenden Datenbeständen der bereits angeführten Eingabemodelle zu.

NP-schwere Probleme sind trotz Verwendung von Search Procedures teilweise überhaupt nicht oder nicht immer in akzeptabler Zeit lösbar. Die Struktur der Optimierungsprobleme eignet sich jedoch zur Parallelisierung, so dass die Verteilung der Berechnungen auf mehrere CPU's ein Weg zur Reduktion der Kosten darstellt. Es wurde vielfach dargelegt, dass insbesondere die Anwendung der Search Procedures entscheidend für die Lösbarkeit eines Optimierungsproblems ist. Die Problematik besteht in der auf den konkreten Einzelfall bezogenen, äußerst schwierigen, jedoch unbedingt notwendigen Definition dieser Prozeduren. Eine Vereinfachung dieser Implementierung ist für das Gesamtkonzept unerlässlich. Eine potentielle Zielrichtung ist hier die Anzeige der für den Optimierungsalgorithmus bedeutenden Variablen zur Zielerfüllung und die daraus abgeleitete automatisierte Erzeugung von Search Procedures.

### **Performancemodell**

Die Verbesserung des Performancemodells ist aus zweierlei Gründen von besonderer Bedeutung. Einerseits ist es möglich, dass aus einer Optimierungsrechnung kein perfektes Ergebnis hervorgebracht werden kann. In diesem Fall muss der Architekt die Eingabeparameter des Optimierungsmodells, resp. die Variablen, Nebenbedingungen und Zielfunktionen entsprechend

verändern und eine erneute Berechnung durchführen. Diese Veränderungen unterliegen jedoch unter Umständen der Unkenntnis, welche Parameter gezielt gesteuert werden müssen, damit ein zufrieden stellendes Ergebnis erzielt werden kann. Andererseits ist es möglich, dass eine Optimierungsrechnung eine große Anzahl von Lösungen hervorbringt, die vom Architekten gesichtet und ausgewertet werden müssen. Die Prototypen zeigen, dass bei Problemen mittlerer und hoher Komplexität, Hunderte oder Tausende von Lösungen erzielt werden können. In beiden Fällen muss das Performancemodell die Einbindung von Szenarien ermöglichen. Diese müssen in Form eines Szenario-Managers die Abhängigkeiten der Variablen, Nebenbedingungen und Zielfunktionen und deren Gewicht für die bereits ermittelte oder nicht ermittelbare Lösung numerisch und graphisch anzeigen. In einem rekursiven Prozess muss die Neu-Gewichtung, Entfernung oder die Addition weiterer Parameter unter Anzeige eines potentiell zu erreichenden Optimierungsergebnisses ermöglicht werden. Eine wichtige Kenngröße der ermittelten Lösungen ist die Vergleichbarkeit mit einer potentiellen Neubaumaßnahme. Zur Beurteilung dieser Gesamtpformance müssen Parameter zur Baukostenermittlung in beide Modelle integriert werden.

Die erfolgreiche Anwendung des Zerstörungssarmen Modells basiert unter anderem auf der Granularität der Netztopologie. Da die Verfeinerung des Netzes eine Vergrößerung der Abstände zwischen den Knoten mit sich bringen kann, ist die Messung der absoluten Kantenlängen sinnvoll. Die Integration einer Funktionalität zur Trennung oder Zusammenführung existenter Knoten kann den Lösungsraum ebenfalls potentiell erweitern. Die veränderbare Granularität repräsentiert hier das Verhältnis von ökologisch sinnvollem und planerisch machbarem. Die mit dem Abbruch oder der Errichtung Raumtrennender Elemente verbundenen Stoffflüsse und Kosten müssen dazu im Performancemodell als Design-Variablen abgebildet werden.

Das Zerstörungsreiche Modell beinhaltet derzeit keine Messung der Performance, da die Prämisse dieses Modells die Erfüllung aller Nebenbedingungen und Zielfunktionen ist. Es ist jedoch gleichermaßen möglich, durch die Einführung gewichteter Zielfunktionen den Lösungsraum zu erweitern und die Performance jeder Lösung zu messen.

## 8.4 Schlussbetrachtung

In der vorliegenden Arbeit wurde ein Systemkonzept entwickelt, welches dem Architekten zur Entscheidungsunterstützung bei Umnutzungs- und Umbauaufgaben dienen kann. Es wurde gezeigt, dass Planungsaufgaben zur Neuordnung einer bestehenden Grundrisskonstellation in Form von Optimierungsproblemen dargestellt und gelöst werden können. Es steht außer Frage, dass allein auf der Grundlage der erzeugten Lösungen keine Entscheidung zu Gunsten oder Ungunsten einer Revitalisierungsmaßnahme getroffen wird. Zweifelsfrei bestimmten viele weitere Faktoren diese Entscheidung. Das entwickelte Systemkonzept ist daher als einer von vielen möglichen Beiträgen zu verstehen, mit dessen Hilfe die Idee der Nachnutzung des Bestandes protegert werden kann. Nicht weil es den Architekten überzeugen würde, dass die Verfolgung dieser Idee wichtig ist, sondern weil es ihm bei der Ermittlung von Lösungen, in einer für ihn schwierigen Domäne mit einer unvorstellbar großen Anzahl möglicher Lösungswege, praktische Unterstützung bietet. Der Architekt entscheidet nach wie vor, ob eine der automatisiert erzeugten Lösungen weiterverfolgt oder umgesetzt wird, er kann diese Entscheidung jedoch auf einer qualitativ bewertbaren Grundlage treffen. Diese Methodik der Abbildung von Planungsaufgaben in Form von Optimierungsproblemen kann auf alle jene Bereiche der Planung angewendet werden, in denen quantifizierbare Ziele oder Bedingungen definiert werden können.

Aus den gewonnenen Erfahrungen bei der Auseinandersetzung mit Optimierungsverfahren und aus den Erfahrungen in der Planungspraxis muss jedoch konstatiert werden, dass kaum ein praktizierender Architekt in der Lage, und voraussichtlich auch nicht Willens wäre, sich in die Methodik der Optimierungsrechnung einzuarbeiten. Neben den alltäglichen Anforderungen die an den Beruf des Architekten gestellt werden, müssten nun noch fundierte Kenntnisse der Mathematik, Informatik und des Operations Research erworben werden. Mit einem derartigen Anspruch anzutreten ist illusorisch.

Wie in der zu Beginn von Pedersen rezipierten griechischen Mythologie liefert nur die Existenz der Gegensätzlichkeiten eine Lösung. Es muss daher Architekten geben, die ihr Berufsbild nicht in dem des traditionellen Entwurfsarchitekten sehen. Diese Architekten benötigen grundlegende Kenntnisse unterschiedlichster Domänen. Sie handeln als Vermittler zwischen den Experten und repräsentieren die Dualität des Bogens und der Lyra.



## **Appendix A**

## A1 – Abkürzungen und Akronyme

AI	Artificial Intelligence
AMPL	Algebraic Modeling Programming Language (Algebraic Modeling Language for Linear and Nonlinear Optimization Problems)
CAAD	Computer Aided Architectural Design
CAD	Computer Aided Design
CHIP	Constraint Handling In Prolog (Constraint Logic Programming Language)
CLP(R)	Constraint Logic Programming (Real)
F2Border	Force to Border (Constraint)
FConn	Force Connection (Constraint)
FInside	Force Inside (Constraint)
FOutside	Force Outside (Constraint)
GA	Genetischer Algorithmus
GAMS	General Algebraic Modeling System
HNF	Hauptnutzfläche (DIN 277)
IFC	Industry Foundation Classes
IP	Integer Programming
KI	Künstliche Intelligenz
LP	Linear Programming
MFLOPS	Million Floating Point Operations Per Second
MILP	Mixed Integer Linear Programming
MIPS	Million Instructions Per Second
NF	Nutzfläche (DIN 277)
NNF	Nebennutzfläche (DIN 277)
NP	nichtdeterministisch polynomial
OPL	Optimization Programming Language

---

PConn	Prohibit Connection (Constraint)
PInter	Prohibit Intersection (Constraint)
Prolog	Programming in Logic
RPA	regelbasiert parametrisch-assoziativ
VF	Verkehrsfläche (DIN 277)
WBS	Wissensbasierte Systeme
XPS	Expert System
ZA	Zerstörungsarm(es Modell)
ZR	Zerstörungsreich(es Modell)

## A2 – Abbildungen

Abb. 2.1	Der deutsche Gebäudebestand im Referenzjahr 1991 in Anlehnung an [Hassler und Kohler, 2000].....	23
Abb. 2.2	Grundrisse eines Universitätsgebäudes und eines ehemaligen Franziskanerklosters mit massiven Tragkonstruktionen [Dolgner und Dolgner, 1996]; [Meyer-Bohe, 1991] .....	24
Abb. 2.3	Grundriss eines Bürohauses in Berlin (1967) in Skelettbauweise [Nagel und Linke, 1969] ....	25
Abb. 2.4	Gebäude mit unflexibler und flexibler Grundrissorganisation [Meyer-Bohe, 1991]; [Nagel und Linke, 1969].....	26
Abb. 2.5	Die gängige Vorgehensweise - Neubauen statt Umdenken.....	28
Abb. 2.6	Baufaufgaben der Zukunft .....	39
Abb. 3.1	Stufenweise, strukturelle Dekomposition um Tertiär- und Sekundär- zur Primärstruktur .....	46
Abb. 3.2	Organisatorische Dekomposition – Einzelflächen / Nutzungseinheiten / Gesamtflächen .....	46
Abb. 3.3	Stufenweise Neuordnung einzelner Flächen aus Abb. 3.2 .....	47
Abb. 3.4	Zeitlicher Ablaufplan des Zerstörungsreichen Modells .....	53
Abb. 3.5	Entwurfsaufgabe (Zerstörungsreiches Modell) - Alternative Lösungen nach gleichen Anforderungen.....	54
Abb. 3.6	CTA Containerterminal, Hamburg Altenwerder, Analogie der Modellbildung.....	55
Abb. 3.7	Unterschiedliche Lösungen eines Perfect Square Problems mit 21 Quadraten.....	55
Abb. 3.8	Zeitlicher Ablaufplan des Zerstörungsarmen Modells.....	59
Abb. 3.9	Entwurfsaufgabe (Zerstörungsarmes Modell) - Alternative Lösungen nach gleichen Anforderungen.....	59
Abb. 3.10	Schach [hjb, 2004]; Go [Sente, 2005].....	60
Abb. 3.11	Methodik des ZR-Modells, Methodik des ZA-Modells.....	61
Abb. 3.12	Objektive und subjektive Entwurfsregeln.....	63
Abb. 3.13	Raster mit 5 x 3 Einheiten, Exemplarische Belegung des Rasters .....	64
Abb. 4.1	Klassischer Aufbau Wissensbasierter Systeme nach [Waterman, 1986].....	67
Abb. 4.2	Semantisches Netz eines Mehrfamilienhauses (MFH) .....	70
Abb. 4.3	Repräsentation von Wissen in WBS .....	71
Abb. 5.1	Bubble-Diagramm eines Grundrisses .....	82
Abb. 5.2	Graphendarstellung.....	82
Abb. 5.3	Grundrissabstraktion – Zerlegung in rechteckige Units.....	84

Abb. 5.4 Geometrische Definition einer Unit mit zwei bzw. vier Abstandsvariablen .....	84
Abb. 5.5 Definition nicht-rechtwinkliger Units .....	85
Abb. 5.6 Void-Unit (grau) in imaginärer rechteckiger Grundunit.....	87
Abb. 5.7 Schematische Darstellung des Optimierungsmodells .....	92
Abb. 5.8 Geometrische Definition der Units $i$ und $j$ innerhalb der Grundunit $R$ .....	94
Abb. 5.9 Repräsentation eines Grundrisses durch einen Graphen .....	106
Abb. 5.10 Schematische Darstellung des Performancemodells.....	109
Abb. 7.1 Konzeptioneller Aufbau CATIA-Prototyp .....	134
Abb. 7.2 CATIA Produktmodell .....	135
Abb. 7.3 Lokales Optimum als Ergebnis des CATIA-Prototyps.....	137
Abb. 7.4 Konzeptioneller Aufbau VisualDOC-Prototyp.....	138
Abb. 7.5 CTMIN 1 - Constraint Erfüllung im Toleranzbereich 1 und vollständige Erfüllung der Zielfunktion .....	167
Abb. 7.6 CTMIN 0.003 - Basierend auf der Berechnung mit dem Best-Design-Point von CTMIN 1 ....	168
Abb. 7.7 SLP-Berechnung - Erfüllung aller Constraints und der Target-Zielfunktion .....	169
Abb. 7.8 Konzeptioneller Aufbau OPL-Prototyp.....	170
Abb. 7.9 Lösungen des Tong-Problems .....	176
Abb. 7.10 Lösungen des Pfefferkorn-Problems.....	178
Abb. 7.11 Ergebnisse der Umfangmaximierung.....	180
Abb. 7.12 Rechteckige Grundrissform (dunkelgraue Fläche) .....	182
Abb. 7.13 Flächenneuordnung ohne Search-Procedure, Eiermann-Bau Apolda, 1.OG.....	183
Abb. 7.14 Flächenneuordnung mit Search-Procedure, Eiermann-Bau Apolda, 1. OG .....	183
Abb. 7.15 Manuelle Entwurfslösung, Eiermann-Bau Apolda, 1. OG [Marquardt und Raabe, 2002].....	183
Abb. 7.16 Eiermann-Bau Apolda [InfAR, 2005].....	184
Abb. 7.17 Bubble-Diagramm des Raumprogramms für das Maculet-Problem .....	187
Abb. 7.18 Berechnungszeit des Maculet-Problems .....	188
Abb. 7.19 Choice Points des Maculet-Problems.....	188
Abb. 7.20 Lösungen des Maculet-Problems .....	189
Abb. 7.21 Berechnungszeit des Maculet-Problems mit Search Procedure .....	192
Abb. 7.22 Choice Points des Maculet-Problems mit Search Procedure .....	193
Abb. 7.23 Exemplarische Lösungen des Maculet-Problems mit Search Procedure (y-Richtung).....	194

Abb. 7.24 Exemplarische Lösungen des Maculet-Problems mit Search Procedure (x, y-Richtung).....	196
Abb. 7.25 Exemplarische Lösungen des Maculet-Problems ohne Größenbeschränkung der VF.....	198
Abb. 7.26 Bubble-Diagramm des Raumprogramms, Eiermann-Bau Apolda, EG.....	201
Abb. 7.27 Rechtwinkliger Grundriss (dunkelgrau), Eiermann-Bau Apolda, EG.....	202
Abb. 7.28 Modellierung einer imaginären rechteckigen Grundrissform durch Void-Units.....	203
Abb. 7.29 Ergebnis 1 der Flächenneuordnung, Eiermann-Bau Apolda, EG.....	203
Abb. 7.30 Ergebnis 2 der Flächenneuordnung, Eiermann-Bau Apolda, EG.....	203
Abb. 7.31 Ergebnis 3 der Flächenneuordnung, Eiermann-Bau Apolda, EG.....	204
Abb. 7.32 Ergebnis 4 der Flächenneuordnung, Eiermann-Bau Apolda, EG.....	204
Abb. 7.33 Manuelle Entwurfslösung, Eiermann-Bau Apolda, EG [Marquardt und Raabe, 2002].....	204
Abb. 7.34 Variante 1 - Maximierung des Flächeninhalts.....	207
Abb. 7.35 Variante 2 - Maximierung des Umfangs (Flächenapproximation).....	208
Abb. 7.36 Ergebnisse der Variante 1 unter Verwendung von einer Search Procedure.....	211
Abb. 7.37 Ergebnisse der Variante 2 unter Verwendung von drei Search Procedures.....	211
Abb. 7.38 Komplexität des Raumprogramms (Model: 7332_2c.mod).....	213
Abb. 7.39 Reduktion der Komplexität durch Verwendung von ConnectionSets.....	214
Abb. 7.40 Berechnungsergebnisse mit Search Procedure aus Variante 2 (7332_2b.mod).....	215
Abb. 7.41 Exemplarische Berechnungsergebnisse ohne fconnectionSet19b.....	215
Abb. 7.42 Exemplarische Berechnungsergebnisse mit fconnectionSet19b.....	216
Abb. 7.43 Grundriss und Darstellung des resultierenden Graphen.....	218
Abb. 7.44 Ergebnisse des Flächentauschs.....	220
Abb. 7.45 Grundriss und Darstellung des resultierenden Graphen.....	221
Abb. 7.46 Ergebnisse 1 und 2 des Flächentauschs.....	222
Abb. 7.47 Ergebnisse 1 bis 6 des Flächentauschs mit einer Unteren Schranke von 89 %.....	225
Abb. 7.48 Grundriss (Fritz-Foerster-Bau, Technische Universität Dresden).....	227
Abb. 7.49 Graphendarstellung.....	227
Abb. 7.50 Adjazenzmatrix.....	228
Abb. 7.51 Lösungen 1-12 von 46.....	230
Abb. 7.52 Annäherung der Lösungen 1-21 an die Nebenbedingungen (sizeRooms).....	233
Abb. 7.53 Graphische Darstellung der Lösungen 1-21.....	234
Abb. 7.54 Graphendarstellung.....	235

---

Abb. 7.55 Berechnungsverlauf und Trendlinie der Güte der Lösungen.....	236
Abb. 7.56 Häufigkeitsverteilung der Güte der Lösungen .....	237
Abb. 7.57 Graphische Darstellung der Lösungen Nr. 1816 und Nr. 2010 .....	238
Abb. 7.58 Performance der 16 besten Lösungen .....	239
Abb. 7.59 Graphische Darstellung der Lösungen 1, 2, 4 und 10 .....	240
Abb. 7.60 Gesamtperformance ohne zusätzliche Straftermgewichtung .....	241
Abb. 7.61 Gesamtperformance mit zusätzlicher Straftermgewichtung .....	242
Abb. 7.62 Soll/Ist-Vergleich in m <sup>2</sup> . Lösung 4 - Beste Gesamtperformance.....	243
Abb. 8.1 Zusammenfassende Übersicht der entwickelten Modelle und Prototypen.....	249

## A3 – Tabellen

Tab. 2.1	Vorgehensmodell für Revitalisierungsprojekte im Büroimmobiliensektor [Schlote, 2000].....	33
Tab. 4.1	Phasenmodelle der Wissensmodellierung abgeleitet aus [Rieger, 2001a].....	67
Tab. 4.2	Formen der Wissenserhebung in Anlehnung an [Herre und Heller, 2001].....	68
Tab. 4.3	Wissensarten in Anlehnung an [Rieger, 2001b] .....	69
Tab. 7.1	Force Inside.....	142
Tab. 7.2	Force Inside BFGS.....	143
Tab. 7.3	Force Inside FR.....	144
Tab. 7.4	Force Inside SUO.....	145
Tab. 7.5	Prohibit Intersection.....	147
Tab. 7.6	Prohibit Intersection BFGS.....	148
Tab. 7.7	Prohibit Intersection FR.....	149
Tab. 7.8	Prohibit Intersection SUO.....	150
Tab. 7.9	Force Connection.....	152
Tab. 7.10	Force Connection BFGS.....	153
Tab. 7.11	Force Connection FR.....	154
Tab. 7.12	Force Connection SUO.....	155
Tab. 7.13	Force Connection Genetic Algorithm.....	156
Tab. 7.14	Force Outside.....	158
Tab. 7.15	Force Outside BFGS.....	159
Tab. 7.16	Force To Border.....	161
Tab. 7.17	Force To Border BFGS.....	162
Tab. 7.18	Prohibit Connection.....	164
Tab. 7.19	Prohibit Connection BFGS.....	165
Tab. 7.20	Ermittlung der Güte einer Lösung.....	228
Tab. 7.21	Lösungen 1-12 sortiert nach Güte.....	229
Tab. 7.22	Ermittlung der Güte und der Abweichung einer Lösung.....	230
Tab. 7.23	Lösungen 1-21 sortiert nach Güte.....	232
Tab. 7.24	Schlechteste Lösungen Nr. 1816 und Nr. 2010 mit einer Güte von 197 und 200.....	238



Tab. 7.25 Lösungen 1-16 sortiert nach Güte.....	238
Tab. 7.26 Gesamtperformance ohne zusätzliche Straftermgewichtung .....	241
Tab. 7.27 Gesamtperformance mit zusätzlicher Straftermgewichtung .....	242

## **Appendix B**

## B1 – Mathematische Nomenklatur

Die Mathematische Nomenklatur listet in alphabetischer Reihenfolge alle in den jeweiligen Modellen verwendeten Variablen und Konstanten. Die bei den Beschreibungen zu verzeichnenden Redundanzen sind auf die unterschiedliche geometrische Ausprägung der mathematischen Modelle zurück zu führen.

Variable	Beschreibung
$A$	Adjazenzmatrix
$A_{ij}^x$	Binäre Variable zur Positionsbestimmung z.B. Unit i links von Unit j
$A_{ji}^x$	Binäre Variable zur Positionsbestimmung z.B. Unit j links von Unit i
$A_{ij}^y$	Binäre Variable zur Positionsbestimmung z.B. Unit i unterhalb von Unit j
$A_{ji}^y$	Binäre Variable zur Positionsbestimmung z.B. Unit j unterhalb von Unit i
akt	Bezeichnet den Wert des aktuell größten Raumes der Gruppe. Von Bedeutung für die Bestimmung der Güte
anz	Speichert die Anzahl der gefundenen Knoten. Von Bedeutung um neue Knoten in die Variable <code>nodes</code> einzufügen.
anzChecked	Prüft ob bei der Breitensuche neue, d.h. noch nicht enthaltene Knoten gefunden wurden.
$\mathbb{B}$	Menge der binären Zahlen
$b_i$	Breite der Unit i
$b_R$	Breite der Grundunit R
drin	Temporäre logische Variable, die speichert ob ein gefundener Knoten schon in der Variable <code>nodes</code> enthalten ist.
$\Delta x_i$	Breite der Unit i
$\Delta y_i$	Länge der Unit i
$E$	Menge der Kanten

<i>East</i>	Variable bezieht sich auf Himmelsrichtung Osten.
$E_i$	Wandabstand in östlicher Richtung vom Referenzpunkt der Unit $i$
$E_R$	Wandabstand in östlicher Richtung vom Referenzpunkt der Grundunit $R$
$\varepsilon$	Abstandsstrecke zweier Units $i, j$
$F_i$	Flächeninhalt der Unit $i$
$F(x)$	Zielfunktion
$f(x)$	Zielfunktion
$G$	Graph
$Group_j$	Gruppe $j$
$groups_j$	Anzahl der Räume innerhalb der Gruppe $j$
$g_i(x)$	Nebenbedingung
$I_n$	Endliche Indexmenge, d.h. die Menge der ausgewählten Räume
$i$	Bezeichnung der Unit $i$
$j$	Bezeichnung der Unit $j$
$label_i$	Gruppenzugehörigkeit des Knotens $i$
$l_i$	Länge der Unit $i$
loe	Logische Variable die speichert, ob eine vom Modell gefundene Lösung eine Lösung des Problems ist.
$l_R$	Länge der Grundunit $R$
$N_i$	Wandabstand in nördlicher Richtung vom Referenzpunkt der Unit $i$
<i>North</i>	Variable bezieht sich auf Himmelsrichtung Norden.

$Node_i$	Knoten $i$
$N_R$	Wandabstand in nördlicher Richtung vom Referenzpunkt der Grundunit R
$nbreOfGroups$	Anzahl der Teilgraphen
$nbreOfNodes$	Anzahl der Knoten
$nbreOfSolution$	Speichert die Anzahl der Lösungen
$nodes$	Bei der Breitensuche werden die gefundenen Knoten in die Variable $nodes$ eingefügt, um zu prüfen, welche Knoten bereits gefunden wurden und welche neu sind.
$pso$	Obere Schranke
$psu$	Untere Schranke
$R$	Grundunit R
$R'$	Grundstücksfläche $R'$
$\mathbb{R}_+$	Menge der positiven reellen Zahlen
$South$	Variable bezieht sich auf Himmelsrichtung Süden.
$S_i$	Wandabstand in südlicher Richtung vom Referenzpunkt der Unit $i$
$S_R$	Wandabstand in südlicher Richtung vom Referenzpunkt der Grundunit R
$s$	Summe der Abweichungen für die Bestimmung der Güte
$sizeGroups_j$	Gesamtgröße der Gruppe $j$
$sizeNodes_i$	Größe des Knoten $i$
$sizeRooms_{jk}$	Größe des Raumes $k$ in Gruppe $j$
$\delta$	Berührungsstrecke zweier Units $i, j$
$U_i$	Umfang der Unit $i$

$u_{ij}$	Binäre Variable zur Positionsbestimmung, z.B. Unit i über / unter / rechts / links Unit j Unit i Süd- / Nord- / West- / Ostseite von Unit j
$V$	Menge der Knoten
$v_{ij}$	Binäre Variable zur Positionsbestimmung, z.B. Unit i über / unter / rechts / links Unit j Unit i Süd- / Nord- / West- / Ostseite von Unit j
$West$	Variable bezieht sich auf Himmelsrichtung Westen.
$W_i$	Wandabstand in westlicher Richtung vom Referenzpunkt der Unit i
$W_n$	Wertebereich der Design-Variablen
$W_R$	Wandabstand in westlicher Richtung vom Referenzpunkt der Grundunit R
$w_1 f_1(x)$	Gewicht der Zielfunktion 1
$X$	Breite der Grundunit R
$X'$	Breite der Grundstücksfläche R'
$x_{Ei}$	Wandposition der Unit i in östlicher Richtung
$x_{ER}$	Wandposition der Grundunit R in östlicher Richtung
$x_i$	Abszisse des Referenzpunktes der Unit i
$x_{iL}$	Untere Schranke (Lower Bound)
$x_{iU}$	Obere Schranke (Upper Bound)
$x_R$	Abszisse des Referenzpunktes der Grundunit R
$x_{R'}$	Abszisse des Referenzpunktes der Grundstücksfläche R'
$x_{WR}$	Wandposition der Grundunit R in westlicher Richtung
$x_{Wi}$	Wandposition der Unit i in westlicher Richtung
$Y$	Länge der Grundunit R

$Y'$	Länge der Grundstücksfläche $R'$
$y_i$	Ordinate des Referenzpunktes der Unit $i$
$y_{Ni}$	Wandposition der Unit $i$ in nördlicher Richtung
$y_{NR}$	Wandposition der Grundunit $R$ in nördlicher Richtung
$y_{Si}$	Wandposition der Unit $i$ in südlicher Richtung
$y_{SR}$	Wandposition der Grundunit $R$ in südlicher Richtung
$y_R$	Ordinate des Referenzpunktes der Grundunit $R$
$y_{R'}$	Ordinate des Referenzpunktes der Grundstücksfläche $R'$
$z_i$	Begrenzung eines Zahlenwertes (z.B.: Länge, Fläche, Umfang)
$\mathbb{Z}_+$	Menge der positiven ganzen Zahlen

## B2 – Exemplarischer Programmcode

Der exemplarische Programmcode listet jeweils eine Lösung für ein Zerstörungsreiches und ein Zerstörungsarmes Modell. Die in Kapitel 7 beschriebenen Beispiele des OPL-Prototyps befinden sich auf der beiliegenden CD.

```
// 7.3.3 Exemplarische Ergebnisse (Zerstörungsreiches Modell)
// 7.3.3.2 Rechtwinklige Gebäudegrundrisse - Beispiel 2a

// file: 7332_2a.mod
// mathematical model: mathematical model 1 (ref. chapter 6.1.3)
// project: evaluation of interior connection sets

// objective: maximization
// this model evaluates the implementation of different connection sets
// the importance of search procedures will be pointed out
// only interior connections apply
// constraints:
// units 1 to 5 have to connect to each other
// units 6 to 8 have to connect to either one of units 1 to 5
// a 5m grid applies to all units
// search procedures:
// the origin of units 1 to 8 is restricted to 0..30m
// steps for incremental movement are restricted to 5m
// solution:                variant 2 + variant c
// solver:                   solver
// constraints:              70
// variables:                32
// failures:                 23.013
// choice points:           23.034
// numerical solving time:   02,12 sec. (solver)
// graphical solving time:  17,33 sec. (solver)

int X =...;
int Y =...;
int n =...;
int d =...;
int a =...;
int b =...;

var int+ x[1..n] in 0..X;
var int+ y[1..n] in 0..Y;
```



```

var int+ dx[1..n] in a..b;
var int+ dy[1..n] in a..b;

struct connection{
    int i;
    int j;
};

struct single{
    int i;
};

//PROBLEM 1: CONNECTION SET BETWEEN CORRIDORS
{connection}fconnectionSet1={<1,2>,<1,3>,<1,4>,<1,5>};
{connection}fconnectionSet2={<2,3>,<2,4>,<2,5>};
{connection}fconnectionSet3={<3,4>,<3,5>};
{connection}fconnectionSet4={<4,5>};

//PROBLEM 2: CONNECTION SET BETWEEN CORRIDORS AND MAIN UNITS
{connection}fconnectionSet6={<6,1>,<6,2>,<6,3>,<6,4>,<6,5>};
{connection}fconnectionSet7={<7,1>,<7,2>,<7,3>,<7,4>,<7,5>};
{connection}fconnectionSet8={<8,1>,<8,2>,<8,3>,<8,4>,<8,5>};

string colors[1..8] =
["blue","blue","blue","blue","blue","lightgrey","lightgrey","lightgrey"];
Board B;

// MAXIMIZATION (VARIANT 1: AREA):
// maximize sum (i in 6..8) (dx[i] * dy[i])

// MAXIMIZATION (VARIANT 2: AREA APPROXIMATION)
maximize sum (i in 6..8) 2.0 * (dx[i] + dy[i])

subject to {

forall (i in 1..n) {
    x[i]>=0;
    y[i]>=0;
    x[i]+dx[i]<=X;
    y[i]+dy[i]<=Y;
};

//PROHIBIT INTERSECTION CONSTRAINT:
forall (i in 1..n-1)

```

```

forall (j in [i+1..n])
  x[i]+dx[i]-x[j]<=0 \ /
  x[j]+dx[j]-x[i]<=0 \ /
  y[i]+dy[i]-y[j]<=0 \ /
  y[j]+dy[j]-y[i]<=0 ;

//FORCE CONNECTION CONSTRAINT:
//PROBLEM 1:

or (c in fconnectionSet1) //for 'fconnectionSet1'
  (y[c.i]+dy[c.i]-y[c.j]<=0 &
   y[c.i]+dy[c.i]-y[c.j]>=0 &
   x[c.i]+dx[c.i]-x[c.j]>=d &
   x[c.j]+dx[c.j]-x[c.i]>=d

  \ / y[c.j]+dy[c.j]-y[c.i]<=0 &
      y[c.j]+dy[c.j]-y[c.i]>=0 &
      x[c.i]+dx[c.i]-x[c.j]>=d &
      x[c.j]+dx[c.j]-x[c.i]>=d

  \ / x[c.i]+dx[c.i]-x[c.j]<=0 &
      x[c.i]+dx[c.i]-x[c.j]>=0 &
      y[c.i]+dy[c.i]-y[c.j]>=d &
      y[c.j]+dy[c.j]-y[c.i]>=d

  \ / x[c.j]+dx[c.j]-x[c.i]<=0 &
      x[c.j]+dx[c.j]-x[c.i]>=0 &
      y[c.i]+dy[c.i]-y[c.j]>=d &
      y[c.j]+dy[c.j]-y[c.i]>=d);

or (c in fconnectionSet2) //for 'fconnectionSet2'
  (y[c.i]+dy[c.i]-y[c.j]<=0 &
   y[c.i]+dy[c.i]-y[c.j]>=0 &
   x[c.i]+dx[c.i]-x[c.j]>=d &
   x[c.j]+dx[c.j]-x[c.i]>=d

  \ / y[c.j]+dy[c.j]-y[c.i]<=0 &
      y[c.j]+dy[c.j]-y[c.i]>=0 &
      x[c.i]+dx[c.i]-x[c.j]>=d &
      x[c.j]+dx[c.j]-x[c.i]>=d

  \ / x[c.i]+dx[c.i]-x[c.j]<=0 &
      x[c.i]+dx[c.i]-x[c.j]>=0 &
      y[c.i]+dy[c.i]-y[c.j]>=d &
      y[c.j]+dy[c.j]-y[c.i]>=d

```

```
\| x[c.j]+dx[c.j]-x[c.i]<=0 &
   x[c.j]+dx[c.j]-x[c.i]>=0 &
   y[c.i]+dy[c.i]-y[c.j]>=d &
   y[c.j]+dy[c.j]-y[c.i]>=d);

or (c in fconnectionSet3) //for 'fconnectionSet3'
   (y[c.i]+dy[c.i]-y[c.j]<=0 &
    y[c.i]+dy[c.i]-y[c.j]>=0 &
    x[c.i]+dx[c.i]-x[c.j]>=d &
    x[c.j]+dx[c.j]-x[c.i]>=d

\| y[c.j]+dy[c.j]-y[c.i]<=0 &
   y[c.j]+dy[c.j]-y[c.i]>=0 &
   x[c.i]+dx[c.i]-x[c.j]>=d &
   x[c.j]+dx[c.j]-x[c.i]>=d

\| x[c.i]+dx[c.i]-x[c.j]<=0 &
   x[c.i]+dx[c.i]-x[c.j]>=0 &
   y[c.i]+dy[c.i]-y[c.j]>=d &
   y[c.j]+dy[c.j]-y[c.i]>=d

\| x[c.j]+dx[c.j]-x[c.i]<=0 &
   x[c.j]+dx[c.j]-x[c.i]>=0 &
   y[c.i]+dy[c.i]-y[c.j]>=d &
   y[c.j]+dy[c.j]-y[c.i]>=d);

or (c in fconnectionSet4) //for 'fconnectionSet4'
   (y[c.i]+dy[c.i]-y[c.j]<=0 &
    y[c.i]+dy[c.i]-y[c.j]>=0 &
    x[c.i]+dx[c.i]-x[c.j]>=d &
    x[c.j]+dx[c.j]-x[c.i]>=d

\| y[c.j]+dy[c.j]-y[c.i]<=0 &
   y[c.j]+dy[c.j]-y[c.i]>=0 &
   x[c.i]+dx[c.i]-x[c.j]>=d &
   x[c.j]+dx[c.j]-x[c.i]>=d

\| x[c.i]+dx[c.i]-x[c.j]<=0 &
   x[c.i]+dx[c.i]-x[c.j]>=0 &
   y[c.i]+dy[c.i]-y[c.j]>=d &
   y[c.j]+dy[c.j]-y[c.i]>=d

\| x[c.j]+dx[c.j]-x[c.i]<=0 &
   x[c.j]+dx[c.j]-x[c.i]>=0 &
```

```

    y[c.i]+dy[c.i]-y[c.j]>=d &
    y[c.j]+dy[c.j]-y[c.i]>=d);

//PROBLEM 2:

or (c in fconnectionSet6)           //for 'fconnectionSet6'
  (y[c.i]+dy[c.i]-y[c.j]<=0 &
   y[c.i]+dy[c.i]-y[c.j]>=0 &
   x[c.i]+dx[c.i]-x[c.j]>=d &
   x[c.j]+dx[c.j]-x[c.i]>=d

  \/  y[c.j]+dy[c.j]-y[c.i]<=0 &
      y[c.j]+dy[c.j]-y[c.i]>=0 &
      x[c.i]+dx[c.i]-x[c.j]>=d &
      x[c.j]+dx[c.j]-x[c.i]>=d

  \/  x[c.i]+dx[c.i]-x[c.j]<=0 &
      x[c.i]+dx[c.i]-x[c.j]>=0 &
      y[c.i]+dy[c.i]-y[c.j]>=d &
      y[c.j]+dy[c.j]-y[c.i]>=d

  \/  x[c.j]+dx[c.j]-x[c.i]<=0 &
      x[c.j]+dx[c.j]-x[c.i]>=0 &
      y[c.i]+dy[c.i]-y[c.j]>=d &
      y[c.j]+dy[c.j]-y[c.i]>=d);

or (c in fconnectionSet7)           //for 'fconnectionSet7'
  (y[c.i]+dy[c.i]-y[c.j]<=0 &
   y[c.i]+dy[c.i]-y[c.j]>=0 &
   x[c.i]+dx[c.i]-x[c.j]>=d &
   x[c.j]+dx[c.j]-x[c.i]>=d

  \/  y[c.j]+dy[c.j]-y[c.i]<=0 &
      y[c.j]+dy[c.j]-y[c.i]>=0 &
      x[c.i]+dx[c.i]-x[c.j]>=d &
      x[c.j]+dx[c.j]-x[c.i]>=d

  \/  x[c.i]+dx[c.i]-x[c.j]<=0 &
      x[c.i]+dx[c.i]-x[c.j]>=0 &
      y[c.i]+dy[c.i]-y[c.j]>=d &
      y[c.j]+dy[c.j]-y[c.i]>=d

  \/  x[c.j]+dx[c.j]-x[c.i]<=0 &
      x[c.j]+dx[c.j]-x[c.i]>=0 &
      y[c.i]+dy[c.i]-y[c.j]>=d &

```

```

        y[c.j]+dy[c.j]-y[c.i]>=d);

or (c in fconnectionSet8)           //for 'fconnectionSet8'
    (y[c.i]+dy[c.i]-y[c.j]<=0 &
     y[c.i]+dy[c.i]-y[c.j]>=0 &
     x[c.i]+dx[c.i]-x[c.j]>=d &
     x[c.j]+dx[c.j]-x[c.i]>=d

\|  y[c.j]+dy[c.j]-y[c.i]<=0 &
     y[c.j]+dy[c.j]-y[c.i]>=0 &
     x[c.i]+dx[c.i]-x[c.j]>=d &
     x[c.j]+dx[c.j]-x[c.i]>=d

\|  x[c.i]+dx[c.i]-x[c.j]<=0 &
     x[c.i]+dx[c.i]-x[c.j]>=0 &
     y[c.i]+dy[c.i]-y[c.j]>=d &
     y[c.j]+dy[c.j]-y[c.i]>=d

\|  x[c.j]+dx[c.j]-x[c.i]<=0 &
     x[c.j]+dx[c.j]-x[c.i]>=0 &
     y[c.i]+dy[c.i]-y[c.j]>=d &
     y[c.j]+dy[c.j]-y[c.i]>=d);

    dx[6] * dy[6] >= 200;
    dx[7] * dy[7] >= 200;
    dx[8] * dy[8] >= 200;
};

search {
    B.rectangle(0,0,X,Y,3);
    forall( i in 1..n) {
        B.filledRectangle(x[i],y[i],dx[i],dy[i],1,colors[i]);
        B.rectangle(x[i],y[i],dx[i],dy[i],1);

// Variant A: (no search method used)

// Variant B: (only corridors)
//forall (i in 1..5) {
// tryall (v in 0..30 : v mod 5 = 0)
//   x[i] = v;
//   tryall (v in 0..30 : v mod 5 = 0)
//     y[i] = v;
//   };
};
};

```

```
// Variant C: (corridors and all units)
forall (i in 1..8) {
  tryall (v in 0..30 : v mod 5 = 0)
    x[i] = v;
    tryall (v in 0..30 : v mod 5 = 0)
      y[i] = v;
    };
};

data {
  X =30;
  Y =30;
  n = 8;
  a = 5;
  b =20;
  d = 5;
};
```

```

// 7.3.4 Exemplarische Ergebnisse (Zerstörungsarmes Modell)
// 7.3.4.1 Rechtwinklige und Nicht-Rechtwinklige Grundrisse und Units - Beispiel 3a, 3b

// file: 7341_3.mod (Model)
// mathematical model: mathematical model 2 (ref. chapter 6.2.1)
// project: finding one group of 8 rooms within a highly complex plan

// objective: constraint satisfaction
// this model searches for a group of rooms within an existing floor plan
// constraints:
// adjacency and size of units
// lower bounds:          none (model 3a); 97 % (model 3b)
// upper bounds:         none (model 3a); 103 % (model 3b)
// solution:             (solver)
// numerical solving time: 845,01 sec. (model 3a); 1.620,00 sec. (model 3b)
// graphical solving time: -
// number of solutions:   46 (model 3a); 381 (model 3b)

//Definition des Graphen
int nbreOfNodes = ...;
range indices 1..nbreOfNodes;
int adjacencyMatrix[indices,indices] = ...;
int sizeNodes[indices] = ...;

//Definition der Gruppen
int nbreOfGroups = ...;
range ngroups 1..nbreOfGroups;
int groups[ngroups] = ...;
int sizeGroups[ngroups] = ...;
int sizeRooms[ngroups,1..groups[1]] = ...;

//Prozentsatz der Gruppengröße der erfüllt sein muss
int psu << "Bitte Prozentsatz für untere Grenze eingeben (z.B. 90):";
int pso << "Bitte Prozentsatz für obere Grenze eingeben (z.B. 110):";

//Definition der Labels
range labels 1..nbreOfGroups+((sum(g in ngroups) groups[g]) < nbreOfNodes)*1;
var labels label[indices];

solve {
  forall(i in ngroups) {
    //Anzahl der Knoten pro Gruppe stimmt
    sum(j in indices) (label[j]=i)=groups[i];
  }
}

```

```
// Gesamtgröße der Knoten einer Gruppe ist innerhalb eines best. Prozentsatzes der
// Gruppengröße, d.h. hier wird die Abweichung nach oben und unten von der
// Gruppengröße definiert
(pso/100.0)*sizeGroups[i] >= sum(k in indices) (label[k]=i)*sizeNodes[k] >=
(psu/100.0)*sizeGroups[i];
};
};

//Jeder Knoten muss einen Nachbarn mit gleichem Label haben!
search {
  forall(g in ngroups : groups[g]>1)
    forall(k in indices) when(label[k]=g) do
      sum(h in indices) (adjacencyMatrix[k,h]<>0 & (label[k]=label[h]))>0;
};
```





```
];

sizeNodes = [17, 12, 57, 24, 10, 18, 11, 8, 33, 12, 14, 14, 77, 44, 9, 12, 10, 7, 2,
2, 9, 4, 12, 16, 11, 16, 9, 31, 31, 18, 15, 42, 15, 26, 28, 13, 25, 16, 67, 6, 6, 6, 6,
10, 22, 179, 99, 16, 28, 15, 9, 44, 12, 58, 2, 9, 34, 12, 39, 12, 13, 28, 217, 16, 15,
13, 27, 10, 21, 12, 7, 22, 23, 11, 14, 16, 56, 16];

//Definition der Gruppenanzahl
nbreOfGroups = 1;

// Definition der Raumanzahl innerhalb der Gruppen
groups = [8];

// Definition der Gruppengrößen
sizeGroups = [146];

// Definition der Raumgrößen
sizeRooms = [
    [50,30,12,12,12,12,10,8]
];
```

```
// file: 7341_3.osc (Script)

Model m("7341_3.mod","7341_3.dat");
int nodes[m.ngroups,m.indices];
int anz[m.ngroups];
int drin;
int loe:=1;
int nbreOfSolution:=0;
int anzChecked:=0;
int s:=0;
int akt;

while(m.nextSolution()) do {
  //Initialisierung
  forall(g in m.ngroups) {
    if(m.groups[g]>1) then {
      anz[g]:=0;
    }
    else {
      anz[g]:=1;
    }
    forall(i in m.indices)
      nodes[g,i]:=0;
  }

  forall(g in m.ngroups : m.groups[g]>1) {
    // Ersten Knoten finden
    forall(i in m.indices) {
      if(m.label[i]=g) then {
        anz[g]:=1;
        nodes[g,anz[g]]:=i;
        //cout << i <<endl;
        break;
      }
    }
    anzChecked:=0;
    // Breitensuche
    while((anzChecked)<>anz[g]) do {
      forall(e in 1..anz[g] : anzChecked<e) {
        //cout << "Dran: " << nodes[g,e] << endl;
        forall(l in m.indices) {
          if(m.adjacencyMatrix[nodes[g,e],l]<>0 & (m.label[l]=g)) then {
            //cout << "Gefunden: " << l << endl;
            drin:=0;
            forall(a in 1..anz[g]) {
```

```

        if(nodes[g,a]=1) then {
            //cout << nodes[g,a] << " , " << l <<endl;
            drin:=1;
            break;
        }
    }
    if(drin=0) then {
        anz[g]:=anz[g]+1;
        nodes[g,anz[g]]:=1;
        //cout << "Hinzugefügt: " << l << " Anzahl: " << anz[g] << endl;
    }
}
anzChecked:=anzChecked+1;
}
}

forall(g in m.ngroups)
    if(anz[g]<>m.groups[g] & m.groups[g]>1) then {
        loe:=0;
    }
    if(loe=1) then {
// Güte der Lösung bestimmen
// Der Wert beschreibt die absolute Summe der Abweichung in Quadratmetern
// in allen Gruppen

        s:=0;
        forall(g in m.ngroups : m.groups[g]>1) {
            forall(v in 1..m.groups[g]) {
                akt := max(ind in 1..m.groups[g] : nodes[g,ind] <> 0)
m.sizeNodes[nodes[g,ind]];
                //cout << akt << endl;
                s:=s+abs(akt - m.sizeRooms[g,v]);
                forall(ind in 1..m.groups[g]) {
                    if (nodes[g,ind] <> 0 & m.sizeNodes[nodes[g,ind]] = akt) then {
                        nodes[g, ind] := 0;
                        break;
                    }
                }
            }
        }
    }
//Loesung ausgeben
nbreOfSolution:=nbreOfSolution+1;
cout << "Lösung gefunden!" << endl;

```

```
forall(h in m.indices) {
    cout << "label[" << h <<"] = " << m.label[h] << endl;
}
cout << "Die Lösung hat die Güte: " << s << endl;
cout << endl;
}
loe:=1;
}

cout << nbreOfSolution << " Lösungen gefunden!";
```

## **Appendix C**

## C1 – Curriculum Vitae

Name	Thorsten Michael Lömker, Doz. M.Sc. Arch.
Anschrift	Humboldtstrasse 27, 99423 Weimar Klingenberger Straße 6, 01187 Dresden
Email	thorsten.loemker@tu-dresden.de
2006	Hochschuldozent  Technische Universität Dresden, Fakultät Architektur CALA - Dozentur für Computeranwendung in der Architektur und Landschaftsarchitektur
2003	Wissenschaftlicher Mitarbeiter  Bauhaus-Universität Weimar, Fakultät Architektur Professur Informatik in der Architektur
2000	Architekt  Lömker [MEDIEN] + Architektur, Frankfurt/Main und Wiesbaden
1995	Design Direktor  Kahlen + Partner Architekten, Aachen
1993	M.Sc. (Master of Science)  University of Dundee, Schottland Duncan of Jordanstone College of Art, School of Architecture
1991	Carl-Duisberg Stipendium, Connecticut, USA  Stipendium im Rahmen des Programms zur Förderung von Führungs- nachwuchskräften der deutschen Wirtschaft
1991	Architekt, EDV-Bbeauftragter  Lahmeyer International, Beratende Ingenieure, Frankfurt/Main
1990	Dipl.-Ing. (FH)  Fachhochschule Hildesheim / Holzminden, Fachbereich Architektur Auszeichnung mit dem Haarmann-Preis und der Haarmann-Medaille als hochschulbester Student
1988	IAESTE-Stipendium  Trainee, Karmi Architects, Tel Aviv, Israel

## C2 – Veröffentlichungen

- 2006 Loemker, T.M., *Non-Destructive Floor Space Relocation with the Aid of a Constraint Programming Language*. In: Proceedings of ECAADE 2006 conference, Communicating Space(s), (accepted paper), Volos, Greece, September 2006
- Loemker, T.M., *Revitalization of Existing Buildings through Sustainable Non-Destructive Floor Space Relocation*. In: Proceedings of GBEN 2006 conference, Global Built Environment: Towards an Integrated Approach for Sustainability, (accepted paper), Preston, UK, September 2006
- Loemker, T.M., *Solving Revitalization-Problems by the Use of a Constraint Programming Language*. In: Proceedings of IKM 2006 conference, International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering, (accepted paper), Weimar, Germany, July 2006
- Loemker, T.M., *Digital Tools for Sustainable Revitalization of Buildings*. In: Proceedings of the International Conference on Urban, Architectural and Technical Aspects of the Renewal of the Countryside IV“, (accepted paper), Bratislava, Slovakia, May 2006
- Lömker, T.M.: Designing with machines, Solving architectural layout planning problems by the use of a constraint programming language and scheduling algorithms, In: Proceedings of ASCCAD 2006, American University of Sharjah, VAE, April 2006
- 2005 Loemker, T.M., *Designing with Machines*. In: Proceedings of the Dresden International Symposium of Architecture 2005, Entwurfslehre eine Suche, (in print), Dresden, Germany, April 2005
- Loemker, T.M., *Virtual Reality in der Architektur* In: Digital Proceedings of VR-Anwenderforum 2005, Fakultät Maschinenwesen, Dresden, Germany, May 2005
- 2004 Loemker, T.M., *Formt CAAD die Architektur? Möglichkeiten der Einflussnahme von CAAD auf Bauwerksgestalt*. In: Proceedings of the Dresden International Symposium of Architecture 2004, Aesthetics and architectural composition, Edited by R. Weber, Dresden, Juni 2004
- Donath D., Loemker T.M., Richter K.: *Plausibility in the Planning Process - Reason and Confidence in the Computer-Aided Design and Planning of Buildings*. In: Automation in Construction, Special Issue, Conference of the Association for Computer Aided Design in Architecture, Edited by G. Proctor, Volume/Issue 13/2, Elsevier B.V., pp. 159-166



- 2003 Loemker, T.M. *Plausibilität im Planungsprozess - Digitale Planungshilfen für die Revitalisierung von Gebäuden*. In: Proceedings of IKM 2003 conference, IKM - Internationales Kolloquium über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen, Weimar, Juni 2003
- 2002 Donath, D.; Loemker, T.M.; Richter, K.: *Plausibility in the Planning Process - Reason and Confidence in the Computer-Aided Design and Planning of Buildings*. In: Proceedings of ACADIA 2002 conference, Thresholds between the virtual and the real, Los Angeles, USA, October 2002
- Loemker, T.M.; Bähring, S.: *Blob*. In: Dr. Dotzler, G.; Computer Art Faszination 2002, CAF-Verlag, Dr. Dotzler Medien Institut, Frankfurt, 2002
- 2001 Donath, D.; Loemker, T.M.; Richter, K.: *Boundary Debates - Extension from Analog to Digital Spaces*. In: John. S. Gero, Scott Chase, Mike Rosemann, Key Centre of Design Computing and Cognition, University of Sydney (ed.):CAADRIA 2001, International Conference of the Association for Computer Aided Architectural Design Research in Asia University of Sydney, NWS Australia 19-21 April, 2001, pp. 241 – 249
- 2000 Donath, D.; Loemker, T.: *Illusion, Frustration and Vision in Computer Aided Project Planning. - A Reflection and Outlook on the Use of Computing in Architecture*. In: Proceedings of ACADIA 2000 conference, Eternity, Infinity and Virtuality, Washington DC, USA, October 2000
- Donath,D.; Loemker,T.M.; Petzold, F.: *New Approaches in Computer Aided Project Planning – Towards the Use of a Modular Building Information System Based on Digital Survey in the Revitalisation Process of Existing Buildings*. In: Kós,J.R.; Barbosa,A.S.; Paraizo,R.C. (eds.): SIGRADI 2000, Gráfica Onida Ltda., Rio de Janeiro, 2000, pp. 212-215
- 1996 Loemker, T.M.: *Von der Skizze zur Animation – Raum für Visionen*. In: Dr. Dotzler, G.; Computer Art Faszination 1996, CAF-Verlag, Dr. Dotzler Medien Institut, Frankfurt, Deutschland, 1996
- Interview: *Visualisierung / Photorealistisches Rendering – VR ist immer noch zu teuer*. Handelsblatt 22/5 - 1996, Verlagsgruppe Handelsblatt, Düsseldorf, Deutschland, 1996
- 1995 Interview: *EDV auf Biegen oder Brechen? – Es ist immer noch kein Platz für VR-Applikationen in Architekturbüros*. Computerwoche 28/95, Computerwoche Verlag GmbH, München, 1995
- 1994 Loemker, T.M.; Oesterreich-Rappaport, R.: *Visualisierung als Entscheidungshilfe*. In: AEC-Report 06/94, Dressler-Verlag, Heidelberg, Deutschland, 1994
- Loemker, T.M.: *Virtuelle Realität oder reale Virtualität? – Architekturvisualisierung heute*. In: Dr. Dotzler, G.; Computer Art Faszination 1994, CAF-Verlag, Dr. Dotzler Medien Institut, Frankfurt, Deutschland, 1994

- 1993 Loemker, T.M.; Schmitt, M.: *Visualisierung technischer Daten mit Hilfe von Computer-Simulationssystemen*. In: LI Aktuell 22/93, Frankfurt, Deutschland, 1993
- 1991 Loemker, T.M.: *Entwurf und Konstruktion eines Autohauses in Braunschweig*. In: Autohaus-Magazin, Autohaus Verlag, Ottobrunn, Deutschland, 1991

### **C3 – Copyright**

In der vorliegenden Arbeit werden geschützte Markennamen genannt. An dieser Stelle werden die Firmen und ihre Produkte zusammenfassend aufgeführt. Bei der Verwendung von Markennamen in der Arbeit wird nicht wiederholt auf das Copyright hingewiesen.

BIGDOT ist ein Markenzeichen der Firma:

Vanderplaats Research & Development Inc.

1767 S. 8th St., Ste. 100

Colorado Springs, CO 80906, USA

CATIA ist ein eingetragenes Markenzeichen der Firma:

Dassault Systemes

9 Quai Marcel Dassault

92150 Suresnes, France

Digital Project ist ein eingetragenes Markenzeichen der Firma:

Gehry Technologies, Inc.

12541-A Beatrice Street

Los Angeles, California 90066, USA

ebay® ist ein eingetragenes Markenzeichen der Firma:

eBay Inc.

2145 Hamilton Avenue

San Jose, California 95125, USA

FloorPlan 3D ist ein eingetragenes Markenzeichen der Firma:

IMSI Deutschland GmbH

Dr.-Jakob-Staudt-Straße 4

D-83052 Bruckmuehl

ILOG OPL Studio, ILOG OPL Development Studio™ und ILOG CPLEX® sind eingetragene Markennamen der Firma:

ILOG, Inc.

1080 Linda Vista Ave.

Mountain View, CA 94043, USA

Pro/ENGINEER® ist ein eingetragenes Markenzeichen der Firma:

PTC, Parametric Technology Corporation

140 Kendrick Street, Needham

MA 02494 USA

VisualDOC ist ein eingetragenes Markenzeichen der Firma:

Vanderplaats Research & Development Inc.

1767 S. 8th St., Ste. 100

Colorado Springs, CO 80906, USA

## C4 – Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle unmissverständlich gekennzeichnet.

Bei der Auswahl und Auswertung folgenden Materials haben mir die nachstehend aufgeführten Personen in der jeweils beschriebenen Weise entgeltlich/unentgeltlich geholfen.

1. -
2. -
3. -

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder anderen Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Ich versichere ehrenwörtlich, dass ich nach bestem Wissen die reine Wahrheit gesagt und nichts verschwiegen habe.

Ort, Datum

Unterschrift

## Literaturverzeichnis

- Alexander, Christopher, Ishikawa, Sara, *et al.* (1977) *A pattern language : towns, buildings, construction*, New York, NY, Oxford University Press.
- Anders, Peter (1999) *Envisioning cyberspace : designing 3-D electronic spaces*, New York, McGraw-Hill.
- Bauwelt (1998) Für die Hochschule, *Bauwelt*, Ausgabe 33.
- Bhatti, M. Asghar (2000) *Practical optimization methods : with mathematica applications*, New York, NY, Springer.
- BMVBW (2000) Bericht der Kommission Wohnungswirtschaftlicher Strukturwandel in den neuen Bundesländern, (zuletzt besucht am: 08.03.2006)
- BMVBW (2001) Leitfaden Nachhaltiges Bauen, Januar 2001, (zuletzt besucht am: 08.03.2006)
- BMVBW (2002) Bundesministerium für Verkehr-, Bau- und Wohnungswesen, Pressemitteilung Nr. 392/02, 10. September 2002, (zuletzt besucht am: 10.09.2002)
- Bruggen, Coosje van (1999) *Frank O. Gehry : Guggenheim Museum Bilbao*, New York, NY, Guggenheim Museum Publ.
- Buchanan, Bruce G. (1993) *Readings in knowledge acquisition and learning : automating the construction and improvement of expert systems*, San Mateo, Calif., Morgan Kaufmann.
- Bundesministerium für Verbraucherschutz, Ernährung und Landwirtschaft (1953) Bundesministerium für Verbraucherschutz, Ernährung und Landwirtschaft, Flurbereinigungsgesetz, BGBl I 1953, 591, § 103a, <http://bundesrecht.juris.de/bundesrecht/flurbg/index.html>, (zuletzt besucht am: 08.03.2006)
- Christiaanse, Kees (2004) Kaisersrot, <http://www.kaisersrot.com/>, (zuletzt besucht am: 08.03.2006)
- Congena (1994) *Zukunftsstrategie Kombi-Büro : Chancen für Architektur und Organisation*, München [u.a.], Callwey [u.a.].
- Corbusier, Le (1924) *Vers une architecture*, Paris, Cres.
- Djahanschah, Sabine (1999) 'Einführung', in: *Umbau: Über die Zukunft des Baubestandes*, Hassler, U., Kohler, N. und Wang, W. (Hrsg.), Tübingen, Berlin, Ernst Wasmuth Verlag, 10-13.
- Dolgnier, Angela und Dolgnier, Dieter (1996) *Die Bauten der Universität Halle im 19. Jahrhundert : ein Beitrag zur deutschen Universitätsbaugeschichte*, Halle, Fliegenkopf Verl.
- Domschke, Wolfgang und Drexl, Andreas (2005) *Einführung in Operations Research*, Berlin [u.a.], Springer.
- Donath, Dirk und Lömker, Thorsten M. (2003a) *Planungssysteme*, [http://infar.architektur.uni-weimar.de/infar/deu/lehre/archiv/semesterentwurf/digital\\_space/ws02/participants\\_final.html](http://infar.architektur.uni-weimar.de/infar/deu/lehre/archiv/semesterentwurf/digital_space/ws02/participants_final.html), (zuletzt besucht am: 08.03.2006)
- Donath, Dirk und Lömker, Thorsten M. (2003b) DFG-Antragstellung, Plausibilität im Planungsprozess. Weimar.

- Donath, Dirk, Lömker, Thorsten M., *et al.* (2001) 'Boundary Debates - Extension from Analog to Digital Spaces', in: *CAADRIA 2001 International Conference of the Association for Computer Aided Architectural Design Research in Asia*, Gero, J., Chase, S. und Rosemann, M. (Hrsg.), Sydney, Australia, 241-249.
- Eastman, Charles (1973) Automated Space Planning, *Artificial Intelligence*, Ausgabe 4.
- Eversberg, Bernhard (1995) Im Zweifel das Bewährte - Quergedanken über Datenbanken, <http://www.allegro-c.de/formate/vitruv.htm>, (zuletzt besucht am: 08.03.2006)
- Friege, Henning (1999) 'Ressourcenpolitik', in: *Umbau: Über die Zukunft des Baubestandes*, Hassler, U., Kohler, N. und Wang, W. (Hrsg.), Tübingen, Berlin, Ernst Wasmuth Verlag, S. 16-23.
- Gauchel, Josef (Hrsg.) (1990). KI-Forschung im Baubereich. Berlin, Ernst u. Sohn, Berlin.
- Gehry (2004) Gehry Technologies, Digital Project, <http://www.gehrytechnologies.com/products.html>, (zuletzt besucht am: 08.03.2006)
- Giedion, Sigfried (1929) *Befreites Wohnen*, Zürich, Füssli.
- Glymph, James, Shelden, Dennis, *et al.* (2002) 'A Parametric Strategy for Freeform Glass Structures Using Quadrilateral Planar Facets', in: *ACADIA 2002, Thresholds. Design, Research, Education and Practice in the Space Between the Physical and the Virtual, Conference Proceedings*, Proctor, G. (Hrsg.), California State Polytechnic University, Pomona, USA, ACADIA, Association for Computer-Aided Design in Architecture, 307-325.
- Gohout, Wolfgang (2004) Operations Research: einige ausgewählte Gebiete der linearen und nicht-linearen Optimierung, München, Oldenbourg.
- Gritzmann, Peter und Brandenberg, René (2003) *Das Geheimnis des kürzesten Weges: ein mathematisches Abenteuer*, Berlin, Springer.
- Gutschow, Konstanty und Zippel, Hermann (1932) Umbau: Fassadenveränderung, Ladeneinbau, Wohnhausumbau, Wohnungsteilung, seitliche Erweiterung, Aufstockung, Zweckveränderung; Planung und Konstruktion; 86 Beispiele mit 392 vergleichenden Ansichten, Grundrissen und Schnitten, Stuttgart, Hoffmann.
- Hassler, Uta (1999) 'Umbau, Sterblichkeit und langfristige Dynamik', in: *Umbau: Über die Zukunft des Baubestandes*, Hassler, U., Kohler, N. und Wang, W. (Hrsg.), Tübingen, Berlin, Ernst Wasmuth Verlag, 39-59.
- Hassler, Uta und Kohler, Niklaus (1998) Umbau, [http://www.ifib.uni-karlsruhe.de/web/ifib\\_dokumente/downloads/Umbau.pdf](http://www.ifib.uni-karlsruhe.de/web/ifib_dokumente/downloads/Umbau.pdf), (zuletzt besucht am: 08.03.2006)
- Hassler, Uta und Kohler, Niklaus (1999) 'Umbau : über die Zukunft des Baubestandes', in, (Hrsg.), Tübingen [u.a.], Wasmuth, 4 - 7.
- Hassler, Uta und Kohler, Niklaus (2000) 'Umbau - Die Zukunft des Bestands', in: *Umnutzungen im Bestand - Neue Zwecke für alte Gebäude*, Wüstenrot-Stiftung (Hrsg.), Stuttgart, Karl Krämer Verlag, 157-165.
- Hearn, Millard Fillmore (2003) *Ideas that shaped buildings*, Cambridge, Mass., MIT Press.



- Herre, Heinrich und Heller, Barbara (2001) Ontologiebasierte Wissenssysteme, <http://www.ontology.uni-leipzig.de/Teaching/Onto-Wiss2-2001-BspMedizin.pdf>, (zuletzt besucht am: 08.03.2006)
- Hesse, Hans Gerd und Korbion, Hermann (1992) Honorarordnung für Architekten und Ingenieure: (HOAI); Kommentar, München, Beck.
- hjb (2004) Schachbrett-Programme unter Linux, <http://www.pro-linux.de/spiele/jpgs/schachbretter/xboard.jpg>, (zuletzt besucht am: 08.03.2006)
- IBM (2006) Product Lifecycle Management CATIA V5, <http://www-306.ibm.com/software/applications/plm/>, (zuletzt besucht am: 08.03.2006)
- ILOG, Inc. (2005) Rapid Development and Deployment of ILOG CPLEX Applications With ILOG OPL Studio, <http://www.ilog.com/products/optimization/archive.cfm#>, (zuletzt besucht am: 08.03.2006)
- InfAR (2005) Eiermann-Bau, Apolda. Weimar, Professur Informatik in der Architektur.
- Insider (1998) Vom Raumprogramm zum Vorentwurf, <http://www.cadinsider.de/docs/Insider/insider1.pdf>, (zuletzt besucht am: 08.03.2006)
- Jessen, Johann und Schneider, Jochem (2000a) 'Umnutzung im Bestand: Städtebau - Programm - Gestalt', in: *Umnutzungen im Bestand: neue Zwecke für alte Gebäude*, Wüstenrot-Stiftung (Hrsg.), Stuttgart, Krämer, 14-43.
- Jessen, Johann und Schneider, Jochem (2000b) 'Umnutzung - Blick über die Grenzen : Praxis und Objekte im europäischen Vergleich', in: *Umnutzungen im Bestand: neue Zwecke für alte Gebäude*, Wüstenrot-Stiftung (Hrsg.), Stuttgart, Krämer, 44-95.
- Jung, Steffen (2002) 'Erarbeitung eines wissensbasierten Systems für die Modellierung von Mehrkomponenten-Mehrphasen-Gleichgewichten', *Institut für Thermische Verfahrenstechnik und Prozessgrundlagen* Halle, Martin-Luther-Universität Halle - Wittenberg.
- Jungnickel, Dieter (2005) *Graphs, networks and algorithms*, Berlin, Springer.
- Kahlert, Claus (1999) 'Energieverbrauch im Bestand', in: *Umbau: Über die Zukunft des Baubestandes*, Hassler, U., Kohler, N. und Wang, W. (Hrsg.), Tübingen, Berlin, Ernst Wasmuth Verlag, 108-114.
- Kamm, Peter (2005) 4D-Strukturbauteile, [http://www.kamm-architekten-ag.ch/web\\_05/web\\_05\\_04\\_00\\_00\\_4d.htm](http://www.kamm-architekten-ag.ch/web_05/web_05_04_00_00_4d.htm), (zuletzt besucht am: 08.03.2006)
- Kirchhoff, Jutta, Jacobs, Bernd, et al. (1996) Kostengünstige Umnutzung aufgegebenen militärischer Einrichtungen für Wohnzwecke, Wohnergänzungseinrichtungen und andere Nutzungen, Stuttgart, Fraunhofer IRB Verlag.
- Kohler, Niklaus (1999) 'Modelle und Lebenszyklus des Gebäudebestandes', in: *Umbau: Über die Zukunft des Baubestandes*, Hassler, U., Kohler, N. und Wang, W. (Hrsg.), Tübingen, Berlin, Ernst Wasmuth Verlag, 24-38.
- Lämmel, Uwe und Cleve, Jürgen (2001) *Lehr- und Übungsbuch künstliche Intelligenz*, München, Fachbuchverlag Leipzig im Carl-Hanser-Verlag.

- Lenart, Mihaly (1991) *Expertensysteme in der Architektur und im Bauwesen*, Basel, Birkhäuser.
- Leppich, Jürgen (1999) 'Aufbau eines Experten- und Informationssystems der Raumfahrt', *Fachbereich Verkehrswesen und angewandte Mechanik, Institut für Luft- und Raumfahrt Berlin*, Technische Universität Berlin.
- Lömker, Thorsten M. (2003) 'Plausibilität im Planungsprozess - Digitale Planungshilfen für die Bebaubarkeit von Grundstücken', in: *Digital proceedings: 16th International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering*, Gürlebeck, K., Hempel, L. und Könke, C. (Hrsg.), Bauhaus-Universität Weimar,
- Lömker, Thorsten M. (2005) 'Designing with Machines', in: *II. Dresdner Internationales Architektur-Symposium, Entwurfslehre - Eine Suche*, Joppien, J. (Hrsg.),
- Lorenz, Peter (2004) *Entwerfen : 25 Architekten - 25 Standpunkte*, Stuttgart, Deutsche Verlags-Anstalt.
- Losch, Siegfried (1999) 'Zukünftige Wohnungs- und Wohnbaulandnachfrage in der Bundesrepublik Deutschland', in: *Umbau: Über die Zukunft des Baubestandes*, Hassler, U., Kohler, N. und Wang, W. (Hrsg.), Tübingen, Berlin, Ernst Wasmuth Verlag, 78-85.
- Löschner, Jan und Menzel, Uwe (1993) *Künstliche Intelligenz: ein Handwörterbuch für Ingenieure*, Düsseldorf, VDI-Verlag.
- Lunze, Jan (1994) *Künstliche Intelligenz für Ingenieure : Methodische Grundlagen und Software-technologie*, München, Oldenbourg.
- Macchi, Giorgio (1999) 'Die neue Werterhaltungspolitik der Bauverwaltungen', in: *Umbau: Über die Zukunft des Baubestandes*, Hassler, U., Kohler, N. und Wang, W. (Hrsg.), Tübingen, Berlin, Ernst Wasmuth Verlag, 150-157.
- Maculet, Robert (1991) 'Représentation des connaissances spatiales (algèbre de Manhattan et raisonnement spatial avec contraintes)' Paris, Université Paris VI.
- Marquardt, Cassandra und Raabe, Stefanie (2002) 'Revitalisierung des Eiermann-Bau's Apolda', *Professur Informatik in der Architektur* Weimar, Bauhaus Universität Weimar.
- Mau, Markus (2002) *Logistik: mit Übungsaufgaben und Lösungen*, Köln, WRW-Verlag.
- Medjdoub, Benachir und Yannou, Bernard (2001) Dynamic space ordering at a topological level in space planning, *Artificial Intelligence in Engineering*, Ausgabe 15, (1).
- Meyer-Bohe, Walter (1991) *Umbauten : Alternativen zum Neubau*, Stuttgart, Dt. Verl.-Anst.
- Michalek, Jeremy J. (2001) 'Interactive Layout Design Optimization', *Optimal Design Laboratory* Michigan, University of Michigan.
- Nagel, Siegfried und Linke, Siegfried (1969) *Verwaltungsbauten*, [Gütersloh], Bertelsmann.
- Nantke, Hans-Jürgen (2002) *Nachhaltige Entwicklung in Deutschland : die Zukunft dauerhaft umweltgerecht gestalten*, Berlin, Schmidt.

- Neufert, Ernst (1936) Bau-Entwurfslehre: Grundlagen, Normen und Vorschriften über Anlage, Bau, Gestaltung, Raumbedarf, Raumbeziehungen, Maße für Gebäude, Räume, Einrichtungen und Geräte mit dem Menschen als Maß und Ziel ; Handbuch für den Baufachmann, Bauherrn, Lehrenden und Lernenden, Berlin, Bauwelt-Verl.
- Pedersen, William (1993) 'The Bow and the Lyre', in: *Kohn Pedersen Fox: Architecture and Urbanism*, James, W. A. (Hrsg.), New York, Rizzoli International Publications,
- Pfefferkorn, Charles E. (1975) A Heuristic Problem Solving Design System for Equipment or Furniture Layouts, *Communications of the ACM*, Ausgabe 18, ((5)).
- Puppe, Frank (1996) Wissensbasierte Diagnose- und Informationssysteme : mit Anwendungen des Expertensystem-Shell-Baukastens D3, Berlin [u.a.], Springer.
- Rasch, Heinz und Rasch, Bodo (1928) Wie bauen? : Materialien und Konstruktionen für industrielle Produktion, Stuttgart, Wedekind.
- Rieger, Bodo (2001a) WBS-01: Einführung, <http://andromeda.oec.uni-osnabrueck.de/uwdwi2/rap3/>, (zuletzt besucht am: 08.03.2006)
- Rieger, Bodo (2001b) Wissenrepräsentation, <http://sansibar.oec.uni-osnabrueck.de/uwdwi2/rap3/wbs02.pdf>, (zuletzt besucht am: 08.03.2006)
- Rieger, Bodo (2002) Autokauf mit Neuronalen Netzen, <http://andromeda.oec.uni-osnabrueck.de/uwdwi2/rap3/uebungen/ueb07/ueb.html>, (zuletzt besucht am: 08.03.2006)
- Rittel, Horst W. J. und Reuter, Wolf D. (1992) *Planen, Entwerfen, Design : ausgewählte Schriften zu Theorie und Methodik*, Stuttgart [u.a.], Kohlhammer.
- Rogers, Richard und Gumuchdjan, Philip (1997) *Cities for a small planet*, London, Faber and Faber.
- Scheidegger, Markus (2005) INO Intensivbehandlungs- Notfall- und Operationszentrum. Bern.
- Schill-Fendl, Monika (2004) Planungsmethoden in der Architektur : Grundlagen von Planungs- und Entwurfsmethoden für Architekten komplexer Aufgabenstellungen in interdisziplinären Gruppen, dargestellt am Bereich Sozial- und Gesundheitsbauten, Norderstedt, Books on Demand.
- Schlote, Andreas (2000) 'Vorgehensmodell für ein Revitalisierungsprojekt', in: *Immobilien-Praxis SPEZIAL*, Schlote, A., Lederer, M.-M. und Lemke, H.-J. (Hrsg.), Berlin, Bauwerk Verlag GmbH, 181-199.
- Schumann, Frank J. (2001) 'Methoden und Werkzeuge zur Integration der kundengerechten Wertgestaltung in die Konzeptphase des Produktentwicklungsprozesses', *Fakultät für Maschinenbau und Verfahrenstechnik* Chemnitz, Technische Universität Chemnitz.
- Sente (2005) Play Go on Mac OS X, <http://www.sente.ch/software/goban/BoardBig.jpg>, (zuletzt besucht am: 08.03.2006)
- Sitte, Camillo, Crasemann Collins, Christiane, et al. (2003) *Der Städte-Bau nach seinen künstlerischen Grundsätzen*, Wien [u.a.], Böhlau.

- Stamm-Teske, Walter und Donath, Dirk (2006) Innovative Wohnbauaspekte, [http://infar.architektur.uni-weimar.de/innovative\\_wohnungsbauaspekte/](http://infar.architektur.uni-weimar.de/innovative_wohnungsbauaspekte/), (zuletzt besucht am: 08.03.2006)
- Steinmann, Frank (1997) Modellbildung und computergestütztes Modellieren in frühen Phasen des architektonischen Entwurfs, Weimar.
- Tasli, Sule (1997) Knowledge Representation for Architectural Systems, IAED 501 Graduate Studio - Commentary Bibliography Series. Ankara, Turkey.
- Tesching, Klaus (1996) Antonio Sant'Elia ( 1888-1916 ), [http://www.kunstwissen.de/fach/f-kuns/a\\_mod/elia0.htm](http://www.kunstwissen.de/fach/f-kuns/a_mod/elia0.htm), (zuletzt besucht am: 08.03.2006)
- Thuy, Nguyen Huu Chau und Schnupp, Peter (1989) *Wissensverarbeitung und Expertensysteme*, München, Oldenbourg.
- Van Hentenryck, Pascal und Lustig, Irvin (1999) *The OPL Optimization Programming Language*, Cambridge, Mass., MIT Press.
- Vanderplaats (2006) Creators of Design Optimization Technologies, <http://www.vrand.com/>, (zuletzt besucht am: 08.03.2006)
- Vitruvius (1960) *The Ten Books On Architecture*, New York.
- Volz, Michael und Deutsches Architektur-Museum (Hrsg.) (1999). Die ökologische Herausforderung in der Architektur: Nachhaltigkeit, Bilanzierung, Normung, Recycling. Tübingen, Wasmuth.
- Wang, Wilfried (1999) 'Einführung', in: *Umbau: Über die Zukunft des Baubestandes*, Hassler, U., Kohler, N. und Wang, W. (Hrsg.), Tübingen, Berlin, Ernst Wasmuth Verlag, 8-9.
- Waterman, Donald A. (1986) *A Guide To Expert Systems*, Reading, Mass., Addison-Wesley.
- Web\_Brockhaus (2006), <http://www.brockhaus.de>, (zuletzt besucht am: 08.03.2006)
- Web\_Wikipedia.de, <http://www.wikipedia.de>, (zuletzt besucht am: 08.03.2006)
- Web\_Wissen.de, <http://www.wissen.de>, (zuletzt besucht am: 08.03.2006)
- Wüstenrot-Stiftung (2000) *Umnutzungen im Bestand : neue Zwecke für alte Gebäude*, Stuttgart, Krämer.

## **Glossar**

Ambiguität	Am bi gu i tät [f. 10nur Sg.] <i>Zweideutigkeit, Doppelsinn</i> [ <i>&lt;lat. ambiguitas in ders. Bed., zu ambiguus "zwischen zweien schwankend", zu ambigere (eigtl. ambi-agere) "zweifeln, schwanken", &lt;ambi "zu beiden Seiten" und agere "treiben"</i> ] [Web_Wissen.de].
API	Application Programming Interface. Schnittstelle, über welche die Interaktion zweier Programme realisiert werden kann (z.B. C++ API).
assoziativ	(spontan) verknüpfend, (Verwandtes) verbindend [Web_Wissen.de]
Backward-Chaining	siehe Rückwärts-Verkettung
Berechnungskosten	siehe Kosten
BinPacking	Eine Klasse kombinatorischer Optimierungsprobleme bei der Verpackung und Verladung von Waren für Transport und Lagerung. Es müssen $n$ Gegenstände in eine möglichst kleine Anzahl von $m$ Behältern gepackt werden. Bekannt sind in der Regel die Gewichte oder Volumina.
Constraint Satisfaction	Ein Optimierungsproblem ohne zu minimierende oder zu maximierende Zielfunktion. Durch die Erfüllung der Nebenbedingungen (Constraints) ist das Optimierungsproblem zufrieden stellend (Satisfaction) gelöst.
deklaratives Wissen	Deklaratives Wissen beinhaltet statische Aspekte und Fakten über Objekte und Beziehungen. In der Informatik versteht man unter einer deklarativen Wissensrepräsentation eine Repräsentation, in der Wissen zwar spezifiziert, die Anwendung oder Herkunft des Wissens jedoch nicht ermittelt wird. In der Architektur ist beispielsweise die Ausweisung der Brutto-Grundfläche ( $BGF = 2000 \text{ m}^2$ ) eines Gebäudes eine deklarative Repräsentation.
Dekomposition	De kom po si ti on <i>f. 1.</i> Zerlegung, Auflösung; [Web_Wissen.de]
Feng-Shui	Feng-Shui ist eine Lehre, die Räume in Harmonie mit ihrer Umgebung zu bringen versucht. Eine wichtige Rolle spielt im Feng Shui neben der Yin- und Yang-Lehre und den nach den Himmelsrichtungen ausgerichteten Acht Trigrammen, auch die Fünf-Elemente-Lehre. Feng Shui ist auch im modernen China noch sehr stark verbreitet, und praktisch jedes größere Bauvorhaben wird von einem Feng-Shui-Meister beraten. Seit einigen Jahren erfahren die Lehren des Feng

	Shui auch zunehmendes Interesse in der westlichen Architektur und Innenarchitektur. [Web_Wikipedia.de]
Flächentausch	Prinzipiell werden zwischen mehreren Parteien Flächen (i.d.R. Geschoss- oder Raumflächen) getauscht, wenn die Anforderungen des jeweiligen Eigentümers an seine eigene, zu tauschende Fläche nicht erfüllt werden können und der Erwerb der getauschten Fläche die Erfüllung dieser Ansprüche ermöglicht.
Forward-Chaining	Siehe Vorwärts-Verkettung
Gebäudetausch	Der Gebäudetausch ist ein erweitertes Modell des Flächentauschs, bei welchem sich die Flächen nicht auf Geschoss- oder Raumflächen, sondern auf ganze Gebäudeflächen beziehen.
Genetische Algorithmen	Genetische Algorithmen (GA) sind Verfahren für Optimierungsberechnungen, welche sich an die biologische Evolution anlehnen. Sie arbeiten mit Populationen von Individuen, die sich in mehreren Generationszyklen durch Fortpflanzung und Vererbung verändern.
Inferenz	In fe renz [f. 10] mittels logischer Schlussfolgerungen erworbenes Wissen [<lat. inferre "beitragen, hineintragen, folgern"] [Web_Wissen.de]
Introspektion	Bei der Introspektion (Selbstbeobachtung) beschreibt der Experte wie er bestimmte Problemstellungen löst und welche Strategien er dabei benutzt.
IuK-Technologien	Informations- und Kommunikations-Technologien
kausal	<b>1.</b> ursächlich, das Verhältnis von Ursache und Wirkung betreffend, auf ihm beruhend, Ggs. akausal <b>2.</b> ( <i>ling.</i> ) begründend, die Ursache, den Grund angehend [Web_Wissen.de]
Knowledge Engineer	Darunter ist eine Person (Wissensingenieur) zu verstehen, deren Aufgabe es ist, das Fachwissen eines Experten in eine geeignete Repräsentation zu überführen, die in Wissensbasierten Systemen verarbeitet werden kann.
kombinatorische Explosion	Unter der Annahme, dass ein Computer innerhalb einer Sekunde eine Million Rechenoperationen durchführen kann, benötigt er für die Berechnung der Wegstrecken $s-z$ eines aus 20 Schichten à 2 Knoten (42 Knoten) bestehenden Graphen ca. 1 Sekunde. Die Verdreifachung der Schichten (60 Schichten / 122 Knoten) führt zu einer Berechnungszeit von 37.000 Jahren. Eine weitere Erhöhung auf 70 Schichten

	(142 Knoten) resultiert in einer Berechnungszeit von 37 Mio. Jahren. Da bei geringer Erhöhung der Knotenanzahl, die potentielle Anzahl der möglichen Wege zwischen $s-z$ regelrecht explodiert, spricht man hier von „kombinatorischer Explosion“. [Gritzmann und Brandenburg, 2003]
Komposition	Kom po si ti on [f. 10] 1 Zusammensetzung, Anordnung 2 Aufbau (eines Bildes, eines literarischen Werkes) 3 Musikstück [<lat. compositio, Gen. -onis, "Zusammenstellung, Zusammensetzung", zu componere, [ komponieren]] [Web_Wissen.de]
Konstruktgitterverfahren	Das Konstruktgitterverfahren basiert auf der von Kelly entwickelten Theorie der personalen Konstrukte. Nach Kelly sind personale Konstrukte eine Möglichkeit die Welt zu strukturieren und Verhaltenskonsequenzen aufzuzeigen. Der Ablauf einer Wissenserhebung nach dem Konstruktgitterverfahren ist so aufgebaut, dass der befragte Experte die zur Strukturierung des Wissens notwendigen Begriffe selbst liefert. So werden bei der Befragung drei verwandte Begriffe vorgegeben, woraufhin der Experte ein Merkmal angeben muss, welches nur von zwei Begriffen geteilt wird. Diese Vorgehensweise wird mehrfach mit wechselnden Begriffen wiederholt. Es erfolgt eine qualitative Auswertung, d.h. der befragende Knowledge Engineer ermittelt die benutzten Begriffe und rekonstruiert die Begriffsstruktur. In einem Konstruktgitter (Repertory Grid) werden unterschiedliche Gewichtungen des Experten bezüglich der abgefragten Eigenschaften (Konstrukte) festgehalten. Die Auswertung des Gitters kann mit Hilfe der agglomerativen hierarchischen Clusteranalyse erfolgen. Das Ergebnis ist eine Beispiel-Hierarchie, die das erworbene terminologische und taxonomische Wissen darstellt. Einer der wesentlichen Nachteile des Konstruktgitterverfahrens ist die Monotonie der Wissenserhebung sowie die Tatsache, dass nur flache Relationen erfasst werden können. Wie bei anderen phänomenologischen Methoden besteht die grundsätzliche Kritik, dass Ergebnisse durch gegenseitig aufeinander wirkende Selbstwahrnehmung verfälscht werden.
Kosten	Unter Kosten wird der Aufwand verstanden, den die Lösung eines Problems verursacht. In der Optimierung versteht man darunter z.B. die Berechnungszeit (Sekunden) oder auch die monetären Kosten (Euro) eines Problems.
Künstliches Neuronales Netz	Ein Künstliches Neuronales Netz (KNN) ist ein in Anleh-



	nung an den Aufbau und die Funktion von Nervenzellen des menschlichen Gehirn entwickeltes Konzept der Informationstechnologie in Form von Soft- oder Hardware.
Lineare Optimierung	auch Lineare Programmierung genannt - Modelle mit einer oder mehreren linearen Zielfunktionen und Nebenbedingungen - Variablen sind (zumeist nichtnegative) reelle Werte (vgl.[Domschke und Drexl, 2005]).
Logistik	[die; griechisch] Bezeichnung für Material- und Informationsprozesse, die der Optimierung der Raumüberwindung und Zeitüberbrückung sowie der Minimierung der Lagerhaltung dienen. [Web_Wissen.de]
Nachhaltigkeit	Durch den Umweltgipfel in Rio de Janeiro im Jahr 1992 geprägter Begriff (engl. Sustainable Development). Bezeichnet, dass eine getroffene Maßnahme einen effektiven und fortwährenden Einfluss auf eine Entwicklung ausübt.
Neuronales Netz	Nachbildung der Vernetzung und Funktion von Nervenzellen in Form von Soft- oder Hardware in der Informationsverarbeitung. [Web_Wissen.de]
Nichtlineare Optimierung	Auch nichtlineare Programmierung genannt. Modelle mit einer nichtlinearen Zielfunktion und/oder mindestens einer nichtlinearen Nebenbedingung (vgl. [Domschke und Drexl, 2005]).
NP-schwer	(Engl.: NP-complete). Bezeichnet schwer zu lösende Optimierungsprobleme.
Operations Research	(Abk. OR). Bezeichnet ein wissenschaftliches Fachgebiet, in welchem versucht wird, mit Hilfe eines Optimierungsmodells ein reales Entscheidungsproblem abzubilden und zu lösen.
per sis tent	<Adj.; Med.; Biol.> anhaltend, dauernd [<lat. persistens, Part. Präs. zu persistere „verharren“]
Phänomenologie	<i>die; -, kMz.</i> 1. Teilgebiet einer wiss. Disziplin, das sich mit der Beschreibung und Klassifikation der Gegenstände des Wissenschaftsgebietes befasst 2. philosophische Lehre von den Erscheinungen der Dinge [Web_Wissen.de]
Protokoll-Analyse	Der Protokoll-Analyse liegt die These zu Grunde, dass Experten wenig meta-kognitives Wissen, d.h. Wissen über eingesetzte Problemlösungsstrategien, besitzen. Das Theoretisieren über eine durch den Experten eingesetzte Strategie basiert demzufolge auf einer verfälschten Darstellung des

	<p>Problemlöse-Verhaltens. Dieser Verfälschung soll mit Hilfe des Laut-Denken-Protokolls in einer Anwendungssituation entgegen gewirkt werden. Bei der Protokollanalyse wird demnach eine Problemlösungssituation simuliert und eine Prozedur „Lauten-Denkens“ eingeleitet, welche von einem Knowledge Engineer im Detail protokolliert wird. Die Analyse der Protokolle erfolgt durch Segmentierung (Unterteilung des Textes nach Sprechpausen) und semantische Analyse (Wortarten als potenzielle Argumente in Operator-Argument-Strukturen). Unvollständige Protokolle werden durch Suche und Inferenz vervollständigt, so dass eine Regelgenerierung auf Basis der Operator-Argument-Strukturen erfolgen kann. Die Nachteile der Protokollanalyse liegen im hohen Aufwand der Durchführung und in der Interpretation der Ergebnisse. So sind Granularitäten feststellbar, die durch die Fähigkeit des Experten einen Sachverhalt zu schildern determiniert werden. So wird bei Protokollanalysen kausales Wissen, d.h. Wissen um die Ursache eines Faktums, oftmals durch das kompilierte Wissen verdeckt.</p>
prozedurales Wissen	<p>Bei der prozeduralen Wissensrepräsentation sind die zur Betrachtung des Wissens erforderlichen Kontrollinformationen bereits in das Wissen eingebettet. Sie beinhaltet dynamische Aspekte und das Wissen, wie Fakten ermittelt werden. Die prozedurale Repräsentation der Brutto-Grundfläche eines Gebäudes würde von Angaben über deren Zusammensetzung aus Konstruktions-Grundfläche und Netto-Grundfläche - sowie deren weiterer Zusammensetzung aus Nutzfläche, Funktionsfläche und Verkehrsfläche – Gebrauch machen.</p>
Revitalisierung	<p>Wiederbelebung eines Gebäudes zum Zweck der Weiternutzung durch organisatorische und/oder bauliche Maßnahmen</p>
Rückwärts-Verkettung	<p>Bei der Rückwärts-Verkettung (Backward-Chaining) wird zunächst die Belegung eines oder mehrerer Zielwerte vorgenommen. Der einzuleitende Prozess sucht daraufhin Regeln oder Wertbelegungen, aus denen die Bestätigung der Zielbehauptung abgeleitet werden kann.</p>
Semantik	<p><i>die; -, kMz. (ling.)</i> Teilgebiet der Linguistik, das sich mit der grundsätzlichen Analyse und Beschreibung der Bedeutung sprachlicher Ausdrücke befasst [Web_Wissen.de]</p>
Solver	<p>Lösungskomponenten (mathematische Algorithmen) in Optimierungsumgebungen.</p>

Struktogramm	[n. –s–e; EDV] graphische Darstellung für Computerprogramme [<Struktur und ...gramm]
Suchstrategien	Tiefensuche (depth-first) - Regeln werden solange weiterverfolgt, bis keine weitere Entscheidung getroffen werden kann. Breitensuche (breadth-first) - Es werden zunächst alle Regeln einer Ebene abgearbeitet, bevor eine Regel in die Tiefe weiterverfolgt wird. Bestensuche (best-first) - Es werden Gewichtung- oder Bewertungskriterien aufgestellt, die über die Suchrichtung entscheiden.
Taxonomie	<i>die</i> ; -, <i>kMz.</i> 1. ( <i>biol.</i> ) Einordnung der Lebewesen in ein Kategoriensystem anhand ihrer natürlichen Beziehungen zueinander, Systematik (3) 2. ( <i>ling.</i> ) Teilgebiet der Linguistik, das sich mit der Segmentierung und Klassifikation sprachlicher Einheiten befasst [Web_Wissen.de]
Topologie	(von griech: <i>topos</i> = Ort, Platz und <i>logos</i> = Lehre, Wissen, Wort)
Unstrukturiertes Interview	Beim unstrukturierten Interview befragt der Knowledge Engineer den Experten spontan über sein Wissen. Diese Befragung kann auch während der Lösung von Testfällen geschehen.
Vastu (Vaastu)	Vastu ist eine alte indische Architekturlehre. Sie findet heute wieder weltweit Verbreitung. Vastu will sogenanntes "ganzheitliches" Wohnen und Leben in Harmonie mit den Gesetzmäßigkeiten der Natur und dem Kosmos bringen. Die altindische Baukunst und Architektur basiert auf dieser Lehre. Vastu ist ein Sanskritwort und bedeutet Natur, Umgebung oder Umwelt. Das davon abgeleitete Wort Vaastu bezieht sich spezifischer auf alle Arten von Gebäuden, also nicht nur auf Tempel (wo Vastu häufig zur Anwendung kommt), sondern auch Wohnhäuser, Dörfer und sogar für Landschaften und Städte. Die altindische Kultur basierte auf den vedischen Schriften. Diese Schriften sind ein Gesamtwerk von Zehntausenden von Sanskrittexten über alle Bereiche der menschlichen Kultur. Sie umfassen philosophische, religiöse und spirituelle Inhalte und beschäftigen sich mit der Perfektion vieler Wissensbereiche. In diesen Schriften findet man auch die so genannte Vastu-shastra (die Wissenschaft über die Architektur). Vastu ist verwandt mit der Geomantie und mit Feng-Shui. Einige Vastu-Anhänger behaupten, Vastu sei der Ursprung von Feng-Shui, was praktisch aber nicht nachweisbar ist, da beide Systeme bis in die Zeit vor der Erfindung der Schrift zurückreichen. Heute

	<p>datieren Historiker die Anfänge der Vastu- und Feng-Shui-Lehre zwischen 5000 und 7000 vor Christus. [Web_Wikipedia.de]</p>
Vorwärts-Verkettung	<p>Bei der Vorwärts-Verkettung (Forward-Chaining) werden die möglichen Konsequenzen vorgegebener Ausgangswerte untersucht. Dem Vorwärts-Verkettungs-Prozess geht daher immer die Eingabe und Festlegung (Asserting) von Ausgangswerten voraus. Dabei arbeitet dieser Prozess mit zwei Listen, die iterativ abgearbeitet werden. Nach dem Hinzufügen von Fakten zur Wissensbasis werden diese Fakten auf die Agenda-Liste gesetzt. Danach sucht der Inferenzmechanismus nach relevanten Regeln, die möglicherweise auf ein Agenda-Faktum angewandt werden können und setzt diese auf die Regelliste (Rule-List). Beide Listen werden abgearbeitet wobei die Agenda-Liste durch das Ausführen anwendbarer Regeln und das Ableiten neuer Fakten aus diesen Regeln erweitert werden kann. Der Prozess ist beendet, wenn beide Listen abgearbeitet und leer sind oder wenn – durch die Angabe einer Zielvorgabe – die Zielvorgabe erfüllt ist.</p>
Dhrystone	<p>Benchmark für Arithmetikanweisungen gemessen in MIPS - Basierend auf dem original C Dhrystone 2.1 von Reinhold P. Weicker, Siemens AG</p>
Whetstone	<p>Benchmark für Fließkommanweisungen gemessen in MFLOPS - Basierend auf dem originalen C Whetstone 2.0 übersetzt aus dem originalen Algol Programm von H. J. Curnow und B. A. Wichman</p>