DIGITAL TOOLS FOR SUSTAINABLE REVITALIZATION OF BUILDINGS

Finding new Utilizations through Destructive and Non-Destructive Floor Space Relocation

THORSTEN M. LOEMKER
Technische Universität Dresden, CALA
Computerapplication in Architecture and Landscape Architecture
BZW, Zellescher Weg 19, 01062 Dresden
thorsten.loemker@tu-dresden.de

Abstract. In 1845 Edgar Allan Poe wrote the poem “The Raven”, an act full of poetry, love, passion, mourning, melancholia and death. In his essay “The Theory of Composition” which was published in 1846 Poe proved that the poem is based on an accurate mathematical description. Not only in literature are structures present that are based on mathematics. In the work of famous musicians, artists or architects like Bach, Escher or Palladio it is evident that the beauty and clarity of their work as well as its traceability has often been reached through the use of intrinsic mathematic coherences. If suchlike structures could be described within architecture, their mathematical abstraction could supplement “The Theory of Composition” of a building. This research focuses on an approach to describe layout principles of existing buildings in the form of mathematical rules. Provided that “design” is in principle a combinatorial problem, i.e. a constraint-based search for an overall optimal solution of a design problem, two exemplary methods will be described to apply new utilizations to existing buildings through the use of these rules.

1. Introduction

Re-Designing a building is a complex task. It can be compared to composing music, writing a poem or creating an object of art. All these activities have in common, that they share artistically principles. Unfortunately these principles make the modus operandi of the design process difficult to generalize. It seems to be impossible to get an answer to the question of how an architect designs a building.

In the following chapter it will be pointed out that it is however possible to determine a modus operandi that suits artistically as well as other premises during architectural design. As an example a brief digression will be undertaken to the theory of composition of Edgar Allan Poe's "The Raven". This method will be adopted to solve architectural layout problems though the application of a constraint programming language (OPL) that will be described in chapter 3. The application to the revitalization of existing buildings will be described in chapter 4.

2. The Theory of Composition

2.1. WHAT IS DESIGN?

Design can be described as creativity, intuition, even accident would be commensurable. Poe predicated suchlike creative processes on "the beauty of mathematics". Their realization followed a modus operandi that is as well transferable to architectural design tasks. The beginning of his work was always determined by the intention to create a poem. He started with the definition of an overall objective, e.g. "beauty". To achieve this, he added specific constraints, e.g. "effects", whose assignment was to fulfill the criteria to meet the objective. His well-known poem "The Raven" consisted of the following objective and constraints (Table 1):

TABLE 1. The Theory of Composition.

Intention:	Create a poem to satisfy the common and critical taste
Objective:	Beauty
Constraints:	Consideration of an effect, to be accessible to the soul (beauty)
	Determination of the poems scale, with regard to the level of excitement to be achieved (100 lines)
	Composition of the effect, through the plot or the key (plot)
	Selection of the key, e.g. mourning (melancholia)
	Elaboration of artistic attractiveness, e.g. (refrain)

Integration of essential consequences, e.g. a refrain leads to staves

Determination of sound characteristics, e.g. achieve monotony but vary its contextual application

Selection of a single word to support sound characteristics, e.g. choose the most resonant vowel (o) and best consonant to be articulated (r): create a word: e.g. (nevermore)

Elaboration of a pretense for plausibility: e.g. let it be spoken by an unreasonable character (a raven)

Selection of the most melancholic item: e.g. the death of a woman loved by a man.

Poe demonstrated that it is in fact possible to determine rules (constraints and objectives) that describe artistically pieces of work in a mathematical manner. This approach will be transferred to architectural design.

2.2. THE MODUS OPERANDI IN ARCHITECTURAL DESIGN

The uses of mathematical descriptions imply the following hypotheses:

- Architectural design is affected by rules. These rules can be of objective and subjective nature. Objective rules are generally accepted. They are defined in legally binding land-use plans or development schemes. They as well contain contextual rules that relate to the surrounding of the proposed building site. Subjective rules rely on the interest of the specific architect. They imply design rules and all aspects that relate to the creative element of the design process (Figure 1). These design rules very much relate to Poe's approach in writing a poem. It is assumed that syntactical forms could be extracted from an architect's specific handwriting, to be compiled into a mathematical language.

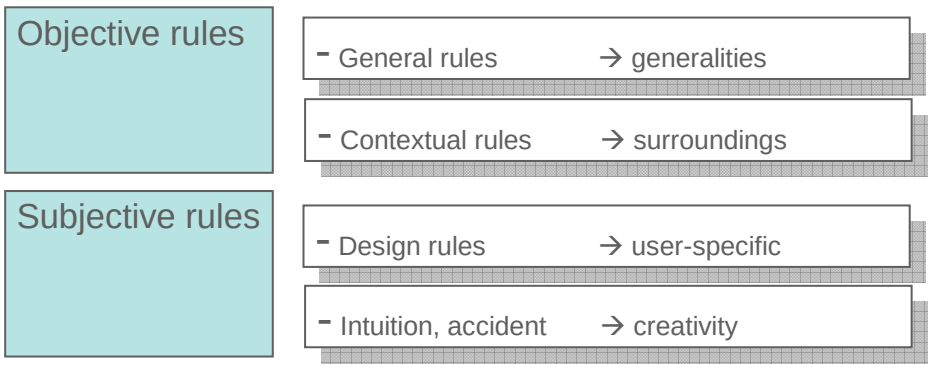


Figure 1. Architectural Design Rules.

- Rules can be used to constrain the solution space of a design problem. Feasible solutions for a design problem can only be found if the search space is constrained. Even simple design tasks can generate millions of solutions. These solutions might not always be feasible in terms of architectural premises.

- Provided that constraints and objectives are specified by the architect, computers can extend the number of feasible solutions for a design problem. Manual design techniques allow the architect to explore a couple of different solutions for a specific design problem. It is hypothesized that the generation of a larger number of feasible solutions is only a question of computing power.

2.3. METHODOLOGY

In the approach presented herein design is a synonym for planning, which could be described as a systematic and methodical course of action for the analysis and solution of current or future problems [Domschke and Drexl, 2005]. The planning task is defined as an analysis of a problem with the aim to prepare optimal decisions by the use of mathematical methods. The decision problem of a planning task is represented by a simulation or optimization model and the application of an efficient algorithm to aid finding one or more solutions to the problem. The basic principle underlying the approach presented herein is the understanding of design in terms of searching for solutions that fulfill specific criteria. In terms of methods used in Operations Research it is possible to classify the above mentioned assumptions as integral components of an optimization model.

The standard form of an optimization model can be expressed as follows [Domschke and Drexl, 2005]:
Minimize or maximize the following objective:

$$z = F(x) \tag{1}$$

Subject to the following constraints:

$$g_i(x) \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} 0 \quad for \;\; i = 1,...,m \tag{2}$$

$$x \in W_1 \times W_2 \times ... \times W_n \;,\; W_j \in \{¡ \; +, ¢ +, B\}, \;\; j = 1,...,n \tag{3}$$

Whereas the symbols have the following meaning:

x	A vector of variables with n components $x_1,\ldots,x_n$
$F(x)$	An objective function
$x_j \in ¡_+$	Nonnegative constraint
$x_j \in ¢_+$	Integer constraint
$x_j \in B_+$	Binary constraint (binary variables)

2.3.1. Objectives:
Formula (1) constitutes the objective function that has to be minimized or maximized. Sometimes it is difficult to define a single objective function that is sufficient to obtain an optimal overall solution for a problem. In this case it is possible to define multi-objective functions whose objectives can be complementary, competing or neutral. In the case of competing objectives a trade-off occurs due to the fact that one objective gets worse whilst others improve. To solve this conflict it is reasonable to assign weights to single objectives (w_1, w_2,…) and to build the weighted sum of the original functions in the following form:

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

No conflicts occur in the case of complementary objective functions due to at least one state that is an optimum for all objective functions. In this case a so called “perfect solution" has been reached. Neutral objective functions do not interfere in their mode of action.

2.3.2. Constraints:
Formula (2) represents a set of equations or inequations which act as restrictions that constrain the search space.

2.3.3. Variables:
Formula (3) defines the domain of variables. These can be continuous, integer or binary. It is important to choose correct variable types within the optimization model. Their range of values is of vital importance for the total cost of computation.

3. Modeling

3.1 OPTIMIZATION MODEL

Architectural objects are specified according to a parametric-associative paradigm. This means that each object which belongs to the model can be accessed and altered by the use of parameters. A room for example consists as an object with geometric parameters such as length, width and height. Objects can as well imply alphanumerical parameters such as their occupancy or materials used. Parameters are defined in the form of variables or constants, whereas variables can be used as inputs for the optimization process. Responses result from the composition of other variables. If a variable is changed during the optimization process dependant variables will be changed as well (Figure 2).

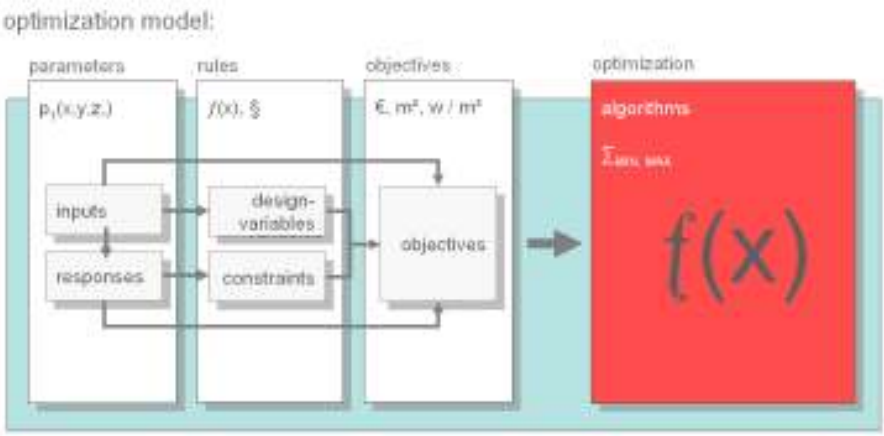


Figure 2. Optimization Modell.

Inputs and Responses are often named Optimization Variables. These variables form the basis of constraints and objective functions. Both must be functions of one or more optimization variables [Bhatti, 2000]. Within an architectural problem domain a response variable could be the area occupied by a specific room. Through multiplication of two input parameters (width and length) a response variable would be rendered. However, it is of primary interest that suchlike parameters generate serious problems for the optimization process due to their non-linear form. Once the design problem is stated in form of design variables, constraints and objectives, the parameters will be passed to the optimization engine which tries to find a feasible solution to the problem.

4. Destructive and Non-Destructive Models

Due to its nature of moving and resizing units, this model alters the plan of an existing building either by breaking down existing walls or by building new walls. Therefore this model is named a “destructive model”.

4.1 DESTRUCTIVE MODEL

The architectural layout problem was formulated with regard to the solution that should be achieved. Rooms were defined as geometric objects with parameters that specified their geometric characteristics. Through the use of topological specifications, relationships between various rooms were defined as constraints. An objective function was implemented that had to be achieved under compliance with the constraints. The following item represents an exemplary problem specification in pseudo-code (Figure 3):

```
// Variable Declaration
        (room 1..n)
// Constant Declaration
        (N)
// Specification of Objectives and Constraints
        maximize, minimize (total area)
                subject to
                  {
                  // Objective Function Specification
                        (total area of room 1..n greater than N)
                  // Constraint Specification
                        (room 1 connects to room 4)
                        (room 5 south of all other rooms)
                        (room 3 greater than room 1 and 5)
                        (rooms 1..n do not intersect with other rooms)
                  }
// Output Parameters
```

Figure3. Exemplary problem specification

It is important to note, that a specification like “room 1 connects to rooms 4” has to be transformed into a mathematical representation. The formulation of an appropriate optimization-model is of utmost importance for a successful evaluation of a design problem solved by a machine. The difficulty for sure exists in the mathematical formulation of the model as well as in the selection of proper algorithms to solve the problem. Even more difficulties arise in the description of usable procedures that classify ordinary design processes.

4.1.1 The Mathematical Model

The principle of the geometric model adopted was the representation of rooms as rectangular units. Michalek (2001) demonstrated this concept in

his work on architectural layout planning. Different from his concept a geometric representation was chosen that describes a rectangular unit through a reference point, a length and a width dimension. Three constraints where taken from this work, that describe the location of a unit inside another (Force Inside), the intersection of two units (Prohibit Intersection) as well as the location of a unit on the border of another unit (Force To Border). Additional constraints were added, that specify the connection of two units (Force Connection), the location of a unit on the outside of another unit (Force Outside) as well as the prohibition of a connection between two units (Prohibit Connection). Various design constraints (e.g. aspect ratio, symmetry) were implemented that refer to subjective rules. These design constraints as well as constraint-combinations make it possible to extend the architects possibility of intervening into the creative process of re-designing a building. The use of constraint-combinations for example, led to a new constraint that made it possible to extend the geometric model to non-rectangular units. These so-called Void-Units approve complex shapes that must not be specified different from other units, according to their geometrical measures. The most important aspect of using OPL as a programming language was its ability to specify search procedures as well as upper- and lower-bounds on variables [van Hentenryck and Lustig, 1999]. The use of search procedures is of uppermost interest due many problems in architectural layout planning that are NP-complete. Their application can dramatically influence the total cost of the optimization process. The mathematical model consists of the following variables, constants and constraints (5-13).

Variables:

Unit i, j	floor space of units
Unit R	floor space of surrounding reference unit
Unit R´	floor space of building site
(I_1, I_2, I_3)	finite-dimensional index sets of rooms
$\varepsilon \geq 0$; $\delta \geq 0$	contact range ; spacing range
(x_i, y_i) ; (x_j, y_j)	non-negative reference points unit i,j
$(\Delta x_i, \Delta y_i)$; $(\Delta x_j, \Delta y_j)$	non-negative width and length unit i,j

$$u_{ij}, v_{ij} \in \{0,1\} \qquad (u_{ij}, v_{ij}) = \begin{Bmatrix} (0,0), \text{ iff. Unit i above Unit j} \\ (0,1), \text{ iff. Unit i under Unit j} \\ (1,0), \text{ iff. Unit i right of Unit j} \\ (1,1), \text{ iff. Unit i left of Unit j} \end{Bmatrix}$$

Variables for Force To Border Constraint:

$$u_{ij}, v_{ij} \in \{0,1\} \qquad (u_{ij}, v_{ij}) = \begin{Bmatrix} (0,0), \text{ iff. Unit i on southside} \\ (0,1), \text{ iff. Unit i on northside} \\ (1,0), \text{ iff. Unit i on westside} \\ (1,1), \text{ iff. Unit i on eastside} \end{Bmatrix}$$

Constants:

(x_R, y_R); $(x_{R'}, y_{R'}) \coloneqq (0,0)$ Non-negative reference point units R, R´

X ; Y ; X' ; Y' Non-negative width and length units R, R´

Force Inside Constraint (FInside)

$$x_i \geq x_j \,;\; y_i \geq y_j \,;\; x_i + \Delta x_i - (x_j + \Delta x_j) \leq 0 \,;\; y_i + \Delta y_i - (y_j + \Delta y_j) \leq 0 \tag{5}$$

Prohibit Intersection Constraint (PInter) (6)

$$\begin{aligned} x_i + \Delta x_i - x_j + u_{ij} X' + v_{ij} X' &\leq 2X' \\ x_j + \Delta x_j - x_i + u_{ij} X' - v_{ij} X' &\leq X' \\ y_i + \Delta y_i - y_j - u_{ij} Y' + v_{ij} Y' &\leq Y' \\ y_j + \Delta y_j - y_i - u_{ij} Y' - v_{ij} Y' &\leq 0 \end{aligned}$$

Force Connection Constraint (FConn) (7)

$$\begin{aligned} y_i + \Delta y_i - y_j - u_{ij} Y' + v_{ij} Y' &\leq Y' \\ y_i + \Delta y_i - y_j + u_{ij} Y' - v_{ij} Y' &\geq -Y' \\ x_i + \Delta x_i - x_j + u_{ij} X' - v_{ij} X' &\geq -X' + \delta \\ x_j + \Delta x_j - x_i + u_{ij} X' - v_{ij} X &\geq -X' + \delta \end{aligned} \;;$$

$$
\begin{aligned}
& y_j + \Delta y_j - y_i - u_{ij} Y' - v_{ij} Y' \ \le 0 \\
& y_j + \Delta y_j - y_i + u_{ij} Y' + v_{ij} Y' \ \ge 0 \\
& x_i + \Delta x_i - x_j + u_{ij} X' + v_{ij} X' \ge 0 + \delta \\
& x_j + \Delta x_j - x_i + u_{ij} X' + v_{ij} X' \ge 0 + \delta
\end{aligned}
$$

$$
\begin{aligned}
& x_i + \Delta x_i - x_j + u_{ij} X' + v_{ij} X' \ \le 2X' \\
& x_i + \Delta x_i - x_j - u_{ij} X' - v_{ij} X' \ \ge -2X' \\
& y_i + \Delta y_i - y_j - u_{ij} Y' - v_{ij} Y' \ \ge -2Y' + \delta \\
& y_j + \Delta y_j - y_i - u_{ij} Y' - v_{ij} Y' \ \ge -2Y' + \delta
\end{aligned}
\quad ;
$$

$$
\begin{aligned}
& x_j + \Delta x_j - x_i + u_{ij} X' - v_{ij} X' \le X' \\
& x_j + \Delta x_j - x_i - u_{ij} X' + v_{ij} X' \ge -X' \\
& y_i + \Delta y_i - y_j - u_{ij} Y' + v_{ij} Y' \ \ge -Y' + \delta \\
& y_j + \Delta y_j - y_i - u_{ij} Y' + v_{ij} Y' \ \ge -Y' + \delta
\end{aligned}
$$

Force To Border Constraint (F2Border) (8)

$$
\begin{aligned}
& x_i + \Delta x_i - (x_j + \Delta x_j) - u_{ij} X' - v_{ij} X' \ge -2X' \\
& y_i + \Delta y_i - (y_j + \Delta y_j) + u_{ij} Y' - v_{ij} Y' \ \ge -Y' \\
& x_i - x_j + u_{ij} X' - v_{ij} X' \ \le X' \\
& y_i - y_j - u_{ij} Y' - v_{ij} Y' \ \le 0
\end{aligned}
$$

Prohibit Connection Constraint (PConn) (9)

$$
\begin{aligned}
& x_i + \Delta x_i - x_j + u_{ij}(X' + \varepsilon) + v_{ij}(X' + \varepsilon) \le 2X' + \varepsilon \\
& x_j + \Delta x_j - x_i + u_{ij}(X' + \varepsilon) - v_{ij}(X' + \varepsilon) \le \ X' \\
& y_i + \Delta y_i - y_j - u_{ij}(Y' + \varepsilon) + v_{ij}(Y' + \varepsilon) \ \le \ Y' \\
& y_j + \Delta y_j - y_i - u_{ij}(Y' + \varepsilon) - v_{ij}(Y' + \varepsilon) \ \le -\varepsilon
\end{aligned}
$$

Force Outside Constraint (FOutside) (10)

$$
\begin{aligned}
& x_i + \Delta x_i - x_j + u_{iR}(X' + \varepsilon) + v_{iR}(X' + \varepsilon) \le 2X' + \varepsilon \\
& x_j + \Delta x_j - x_i + u_{iR}(X' + \varepsilon) - v_{iR}(X' + \varepsilon) \le \ X'
\end{aligned}
$$

$$y_i + \Delta y_i - y_j - u_{iR}(Y'+\varepsilon) + v_{iR}(Y'+\varepsilon) \leq Y'$$

$$y_j + \Delta y_j - y_i - u_{iR}(Y'+\varepsilon) - v_{iR}(Y'+\varepsilon) \leq -\varepsilon$$

$$\varepsilon \geq 0$$

Design Constraints (Length Constraint) (11)

$$\Delta y_i \begin{cases} \leq z_i, \text{ upper boundary of Unit i} \\ = z_i, \text{ exact boundary of Unit i} \\ \geq z_i, \text{ lower boundary of Unit i} \end{cases}$$

Design Constraints (Area Constraint) (12)

$$\mathrm{F}_i \begin{cases} \leq z_i, \text{ upper area of Unit i} \\ = z_i, \text{ exact area of Unit i} \\ \geq z_i, \text{ lower area of Unit i} \end{cases}$$

Design Constraints (Perimeter Constraint) (13)

$$U_i \begin{cases} \leq z_i, \text{ upper perimeter of Unit i} \\ = z_i, \text{ exact perimeter of Unit i} \\ \geq z_i, \text{ lower perimeter of Unit i} \end{cases}$$

$$(z_i \in ¡ \ _+ \text{ fixed}, \ \mathrm{i} \in \{1,...,\mathrm{n}\})$$

4.1.2 Results

The surrounding reference unit in the destructive model was organized in a way, that it represents the area of an existing part of a building that has to be redeveloped. The following images show an exemplary layout of 9 units within a quadratic plan. The following constraints and objective had to be fulfilled (Figure 4):

Topological Constraints:
Force Connection between the red and blue unit, red and yellow unit, red and orange unit.
Force Connection between the yellow and midnight-blue unit.
Prohibit Intersection between all units.
Force all units to stay inside the square.

Length Constraints:

Width and length of the red and blue unit must be greater than or equal to 2m and less than or equal to 4m.
Width and length of the orange unit must be 3m.
Width and length of the midnight-blue, Indian-red and yellow unit must be greater than or equal to 2m.

Ratio Constraints:
Red and blue unit must be symmetrical to the yellow unit.
Midnight-blue and Indian-red unit must be symmetrical to the yellow unit.

Objective: Maximize total perimeter of all units.

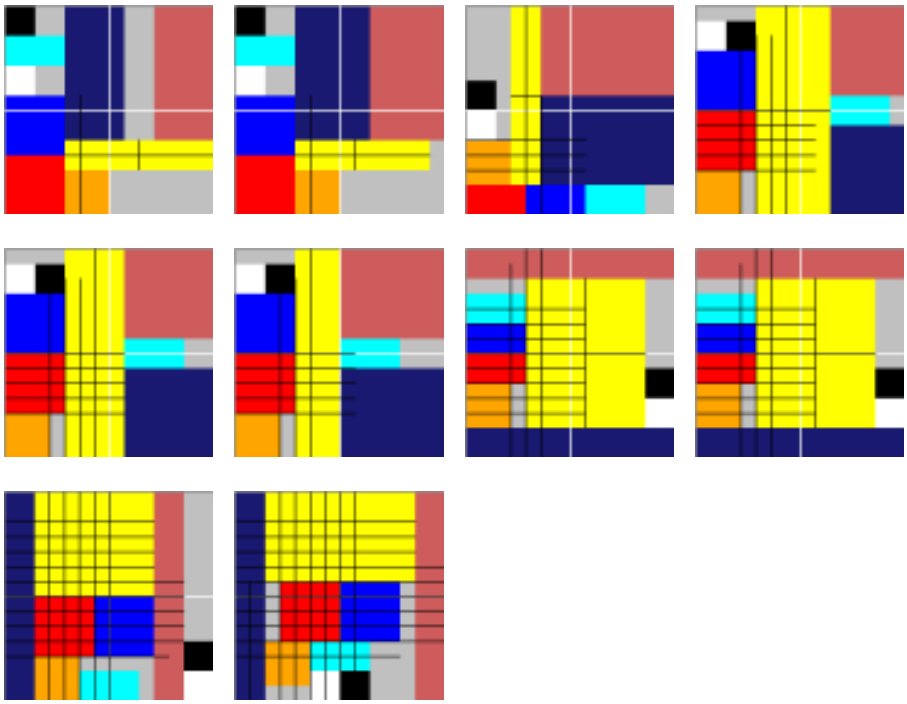


Figure 4. Perimeter optimization of rectangular floor plans

The model calculates 10 different floor plans with a total perimeter of 120 to 144 meters. It reaches an optimum after a couple of minutes. The perimeter objective has been chosen to avoid a non-linear form of the model. This perimeter objective represents an approximation of an area objective which is more common in the architectural practice. Other optimization runs with non-linear constraints showed however, that is as well possible to make use of quadratic representations, particularly if search-methods are implemented in the model.

The next example demonstrates the use of non-rectangular floor plans through the implementation of Void-Units. Constraints were set up through the use of a bubble-diagram (Figure 5). The use of search-methods allowed the solver to calculate the following results in less than 1 second up to 6 seconds (Figure 6).

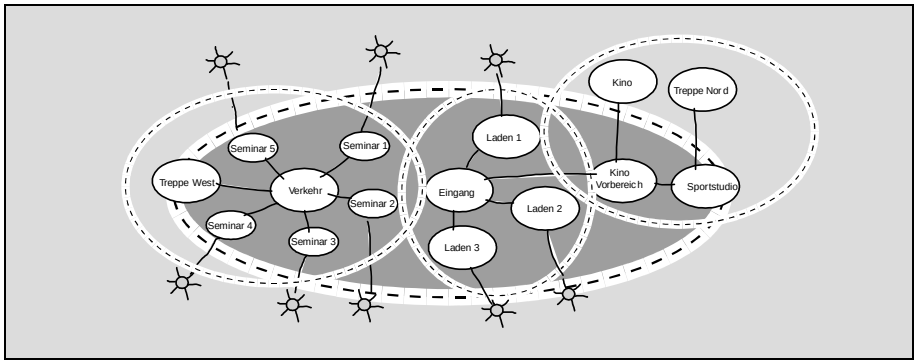


Figure 5. Bubble diagram showing constraints

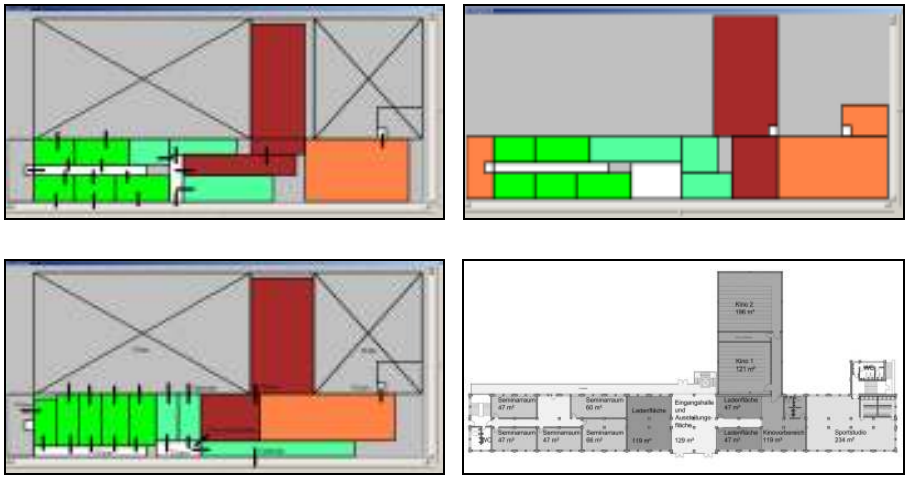


Figure 6. The Results of the optimization run compared to an architects manual design work.

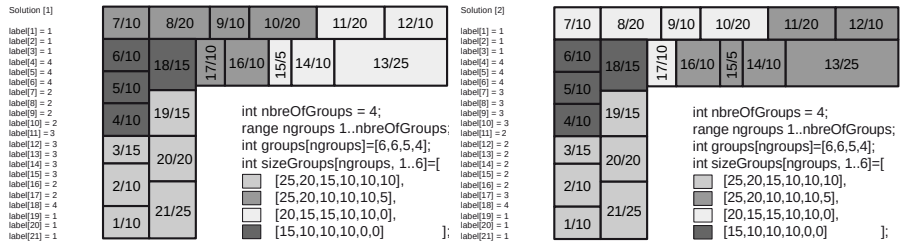
4.2. NON-DESTRUCTIVE MODEL

The revitalization of existing buildings gets more and more important. We are facing a development, where we have to state, that in many cases there is no need to design new buildings due to the fact that an increasing number of existing buildings is not used anymore. In contrast to the method used in the destructive model the most ecological procedure to revitalize these buildings

would be through a continuous usage without making ‘any’ alterations to the stock. Consequentially the model developed for this purpose is a “non-destructive model”. The design problem is related to problems in logistics, e.g. the loading in trans-shipment centers. Our model consists of an adjacency matrix that describes the neighborhood of existing rooms in the stock. Additionally an array stores values of the area of each room. Through the use OPL we defined a procedure that searches for coherent graphs within the matrix. These graphs represent units (groups) of different usage, e.g. *n* office spaces consisting of *x* rooms. The rooms within such a group must represent a coherent graph, which means that each room has to have at least one neighbor. In addition, all members of a group have to fulfill specific requirements regarding the size of each single room and the total number of rooms within the group. These conditions are fulfilled, if the size of each single member of a group is at least the size of an equivalent room that is specified in an array that contains the desired room-sizes. In existing buildings however, this is hardly ever the case. Due to this, the conditions are also fulfilled, if the total size of a group is equal or greater than the total size of the rooms specified in this array. In either case the total number of rooms in a group found has to equal the number of rooms specified as a search criterion. The algorithm searches for equivalent room-sizes first, before it passes over to check group-sizes.

4.2.1 Results

The results of the model developed are encouraging. The images show 8 exemplary assemblies of 21 rooms (Figure 7). All solutions fulfill the requirements made to the size of the rooms, their adjacency and the number of members in a unit (group).



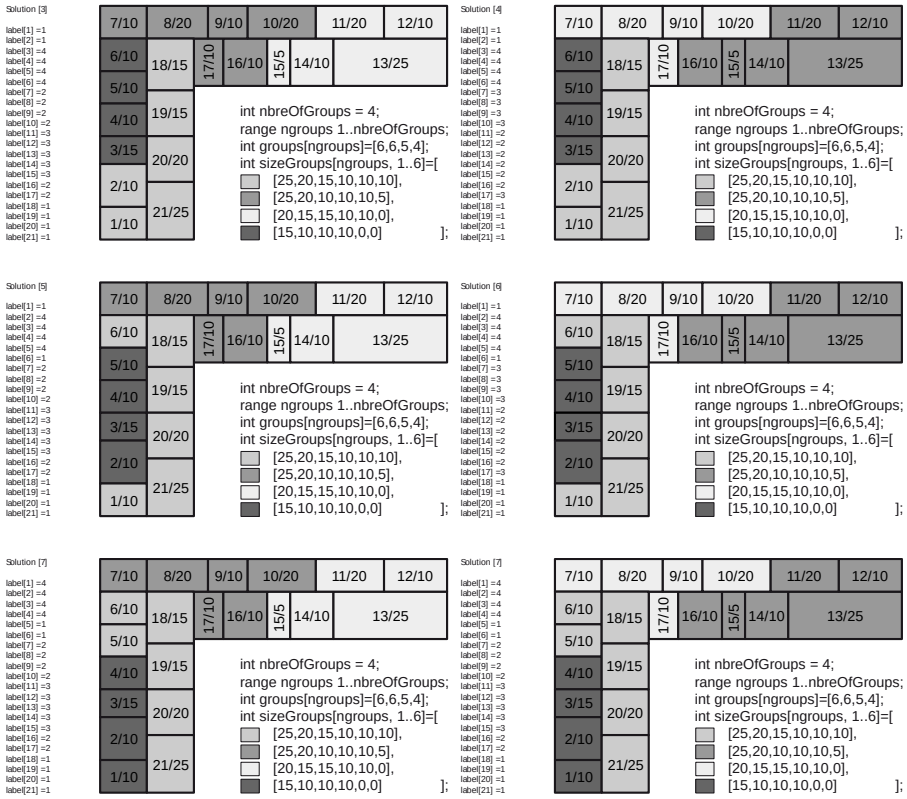


Figure 7. Non-destructive optimization of existing floor plans

We began working with larger models consisting of 78 rooms, respectively 6.084 entries in the matrix. These optimization runs can be solved within 2 hours time on ordinary machines with a reasonable amount of memory. Our latest attempts deal with more than 300 rooms (respectively 90.000 entries in the matrix) arranged on different stories. These tasks are difficult to solve from a computational point of view and might call for parallelization of the program developed.

5. Conclusion

Destructive and especially non-destructive models can play an important role in the revitalization process. Our examinations demonstrate that destructive models which are often NP-complete dramatically benefit from the search methods the user is able to define in OPL. The examination of different models developed, also demonstrates that many models were not

possible to be solved within an appropriate time range without the use of these search methods.

The non-destructive model description differs fundamentally from the description of the mathematical models. By the use of the constraint programming paradigm it is possible to write descriptions of extremely complex tasks within a few lines of code. But not only from a computational point of view are the results of this model promising. In the perpetual important domain of revitalization, destructive and especially non-destructive optimization models can aid the architect in finding answers to design and utilization problems.

References

BHATTI, M. ASGHAR (2000). Practical optimization methods : with mathematica applications. New York, NY [u.a.], Springer.

DOMSCHKE, W. AND DREXL, A. (2005). Einführung in Operations Research : mit 63 Tabellen. Berlin [u.a.], Springer.

MICHALEK, J. (2001). 'Interactive Layout Design Optimization'. Optimal Design Laboratory. Michigan, University of Michigan: 115.

VAN HENTENRYCK, P. AND LUSTIG, I. (1999). The OPL Optimization Programming Language. Cambridge, Mass [u.a.], MIT Press.