Non-Destructive Floor Space Relocation with the Aid of a Constraint Programming Language

Thorsten M. Lömker
Technische Universität Dresden, Germany
http://www.arch.tu-dresden.de/cala/

This research focuses on an approach to describe principles in non-destructive floor space relocation within the domain of revitalization. With the aid of mathematical rules, which are executed by the use of a computer, solutions to floor space relocation problems are generated. Provided that "design" is in principle a combinatorial problem, i.e., a constraint-based search for an overall optimal solution, an exemplary method is described to solve such problems.

Keywords: *Revitalization; Optimization; Constraint Programming; OPL.*

Introduction

Future building tasks will be focused on the examination of existent architecture. Foregoing the building of new structures in favor of reuse and conversion is necessary for economic, ecological, and social reasons. If existing stock does not need to be completely stripped, both procedures are a better alternative to building new structures. Revitalization strategies such as unit swapping, which modifies a building only on the organizational level, are not commonly part of the architect's skill set. Under the premise of sustainability in treating existing buildings, continued use must be achieved with maximal conservation of its structures and minimal changes of its architectural condition. Therefore, this strategy has extraordinary potential. Its high value is based on its ability to be applied to units of arbitrary size and on the use of methods from Operations Research (Domschke and Drexl, 2005) and optimization techniques for solving such problems (Bhatti, 2000).

Methodology

The essential criterion for deciding to revitalize a building is the satisfaction of the room program. In order to attain this, the procedure quickly leads to revitalization solutions that change the building through massive structural modifications. However, given the premise of sustainable treatment of existing buildings, decisions for the continued use of a building should be reached by taking into account the greatest possible conservation of present structures and the least possible alterations of their architectural state. This course of action has found little consideration in previous revitalization efforts. A hypothesis is proposed, stating that the comparison of the room program with the floor plans of a building essentially is a combinatorial problem. Under this assumption it is examined whether solutions for conversion and reuse tasks can be produced automatically by the use of optimization processes in floor plan design. These solutions shall be produced by reordering or swapping of existing areas. The objective is to obtain feasible planning solutions by means

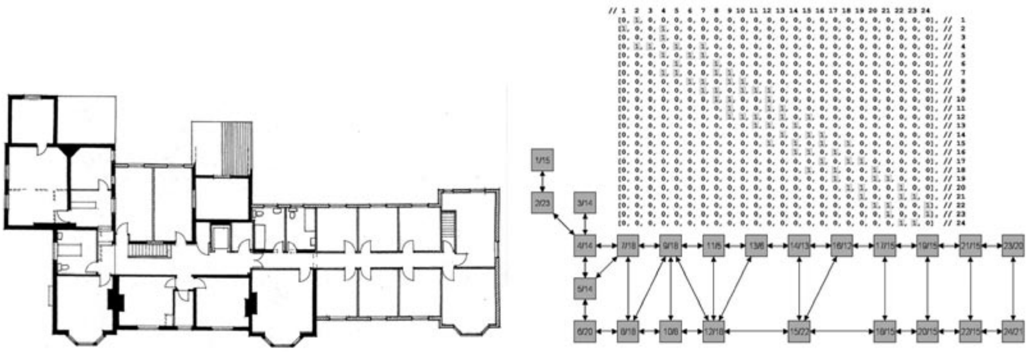


Figure 1
Plan and graph representation

of these computer-based processes, which will serve the architect as a basis for the further editing of the plans.

Non-Destructive Floor Space Relocation

Non-Destructive Models map the reuse strategy, that is, the attempt to reuse the existent room structure of a building while mostly refraining from the removal of walls and other structural elements. The main criterion for this model are the existing structures between room units, for which is determined whether they agree with the structures of a planned future use. This model is mainly applied to buildings with non-separable primary, secondary, and tertiary structures that were built predominantly using massive construction methods. The areas used by the Non-Destructive Model are assembled from existent rooms. It is possible, albeit not desirable, to apply the model to even smaller units, for example, areas decomposed by a grid. This model aims at finding areas in the existent floor plans that satisfy the requirements of the room plan. This is achieved by comparing the properties of the existing areas with the properties of the areas in the room plan. No structural alterations are performed on the existing building. This is possible because the Non-Destructive Model does not change the geometric shape of a room but merely its use profile.

Analogies

The methodology of the Non-Destructive Model is similar to the structure of board games such as Chess, Go, or Connect Four. These games are based on a number of fields or points on a grid, which are occupied by pieces according to certain conditions (game rules). The objective is clearly defined, as it is for the Non-Destructive Model. Both share similar features such as a fixed grid, a restricted possibility to occupy fields, and the satisfaction of a higher objective, that is, to win the game.

Mathematical Model

The Non-Destructive Model focuses on the geometric shapes of existing room limits or the orientation of system lines, such as those of the static system or the grid units of structural components. This abstraction of floor plans ensures non-invasive treatment of existing buildings. Units of the Non-Destructive Model are not limited in their geometric shape. As nodes of a graph they can represent any shape, including non-rectangular shapes.

Conventions:

$G = (V, E)$	graph with
V	set of nodes
$E \subseteq V \times V$	set of edges
$nbreOfNodes$	number of nodes $(=\lvert V \rvert)$
$Node_i$	node i of graph $(i \in \{1,\ldots,\lvert V \rvert\})$
$nbreOfGroups$	number of connected groups (sub graphs) to be found
$Group_j$	group j
$label_{ij}$	means that $label_i = j$

Variables:

$A = a_{ij}$	adjacency matrix of G with $a_{ij} = \begin{cases} 1, \text{ if } (i,j) \in E \\ 0, else \end{cases}$;
$sizeNodes_i$	size of node i in square meters
$groups_j$	number of required units in group j
$sizeGroups_j$	found total size of group j in m²
$sizeRooms_{jk}$	size of room k in group j $(k \in \{1,\ldots,groups_j\})$
psu	lower limit: percentage of required group size that must at least be met (usually $\leq$ 100%)
pso	upper limit: percentage of required group size that must at most be met (usually $\geq$ 100%)
$label_i$	group to which node i belongs

Number:

$$\sum_{i=i}^{|V|} label_{ij} = groups_j \quad \forall j \in \{1,\ldots,nbreOfGroups\} \tag{1}$$

Size:

$$\frac{pso}{100} * sizeGroups_j \geq \sum_{i=i}^{|V|} label_{ij} * sizeNodes_i \geq \frac{psu}{100} * sizeGroups_j \quad \forall j \in \{1,\ldots,nbreOfGroups\} \tag{2}$$

Example

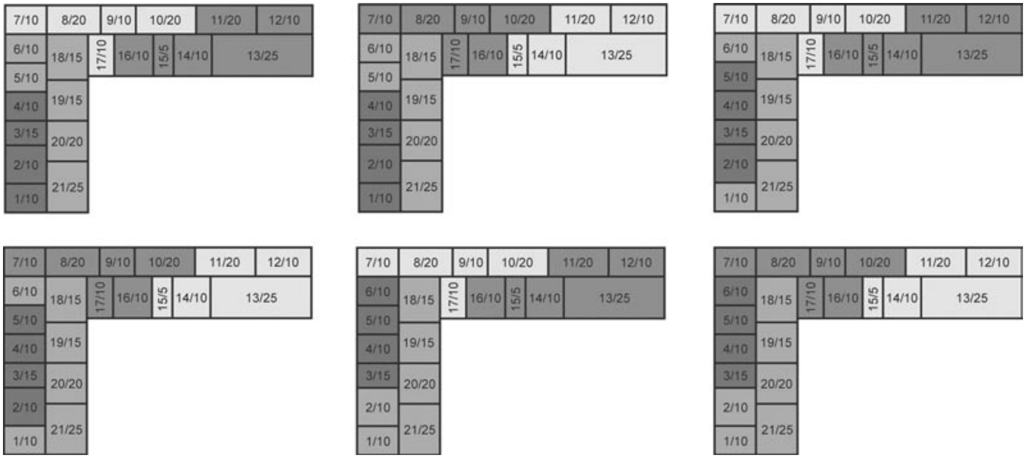
Let there be given a floor plan with 24 rooms and a total area of 369 square meters. Within this floor plan, find 4 room groups (RG) that contain 7, 6, 6, and 5 adjacent rooms (R) whose respective room group area in square meters equals the room group areas (RG1, ..., RG4; ∑ = 369 square meters) required by the room program. Use a Lower Bound if no perfect solution exists.

A preliminary calculation of this example confirms that there is no solution satisfying all auxiliary conditions. The application of a lower bound of 89% produces 6 solutions within 94 seconds. This lower bound defines the largest deviation of an individual unit from the target function (room group size). The total size of the groups, i.e., their distribution on the entire floor plan, is always satisfied by the use of lower bounds. However, since an exact allocation of the existing rooms to the given group sizes is not possible in this example, the prototype tries to determine the constellation having the smallest deviation. For this purpose, some groups are initialized with larger values than those required, others groups with smaller values. This results in an approximation of the values defined by the user.

Conclusion

With the aid of the model developed and the language used (Van Hentenryck & Lustig, 1999), it is possible to represent two- and three-dimensional spatial structures. A two-dimensional matrix can be applied to one or several stories, which can even be located in several buildings. Entire buildings or real estates, including stories, can be represented by block matrices. Using the developed prototype, extremely complex architectural reuse problems have been solved. Here, "extremely complex" refers to problems that could not be solved by an architect in a reasonable amount of time. Furthermore, the prototype can be used to generate all perfect solutions to an optimization problem, which beyond

Figure 2
Solutions of the optimization run with a lower bound of 89%



doubt can not be worked out by an architect. If the target functions can not be completely satisfied, the prototype provides weighted target functions which permit the generation of approximate solutions. The number of generatable solutions depends on the granularity of the network topology. The finer the network, i.e., the higher the number of edges connecting the nodes, the higher the probability of reaching a perfect solution. The performance of the generatable solutions is therefore highly dependent on the architect's readiness to refine the network. Although this refining of the network may increase the distance between room units in a generated solution, it is still recommendable in light of the premise of resource-preserving treatment of building stock. The performance of generated solutions is automatically measured.

References

Bhatti M. A.: 2000, Practical optimization methods : with mathematica applications, Springer, New York, NY.

Domschke W. & Drexl, A.: 2005, Einführung in Operations Research, Springer, Berlin [u.a.].

Van Hentenryck P. & Lustig I.: 1999, The OPL Optimization Programming Language, MIT Press, Cambridge, Mass.