Hospital layout design renovation as a Quadratic Assignment Problem with geodesic distances

Cubukcuoglu, Cemre; Nourian, Pirouz; Tasgetiren, M. Fatih; Sariyildiz, I. Sevil; Azadi, Shervin

DOI
10.1016/j.jobe.2021.102952

Publication date
2021

Document Version
Final published version

Published in
Journal of Building Engineering

Citation (APA)
Cubukcuoglu, C., Nourian, P., Tasgetiren, M. F., Sariyildiz, I. S., & Azadi, S. (2021). Hospital layout design renovation as a Quadratic Assignment Problem with geodesic distances. *Journal of Building Engineering*, *44*, Article 102952. https://doi.org/10.1016/j.jobe.2021.102952

Journal of Building Engineering 44 (2021) 102952

Contents lists available at ScienceDirect

Journal of Building Engineering

journal homepage: www.elsevier.com/locate/jobe

Hospital layout design renovation as a Quadratic Assignment Problem with geodesic distances

Cemre Cubukcuoglu [a,b,*], Pirouz Nourian [a], M. Fatih Tasgetiren [c], I. Sevil Sariyildiz [a], Shervin Azadi [a]

[a] Department of Architectural Engineering and Technology, Chair of Design Informatics, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, the Netherlands
[b] Department of Interior Architecture and Environmental Design, Faculty of Architecture, Yasar University, Izmir, Turkey
[c] Department of International Logistics Management, Faculty of Business, Yasar University, Izmir, Turkey

ARTICLE INFO

Keywords:
Computational design
Architectural space planning
Hospital layout
Quadratic assignment problem
Layout optimization tool

ABSTRACT

Hospital facilities are known as functionally complex buildings. There are usually configurational problems that lead to inefficient transportation processes for patients, medical staff, and/or logistics of materials. The Quadratic Assignment Problem (QAP) is a well-known problem in the field of Operations Research from the category of the facility's location/allocation problems. However, it has rarely been utilized in architectural design practice. This paper presents a formulation of such logistics issues as a QAP for space planning processes aimed at renovation of existing hospitals, a heuristic QAP solver developed in a CAD environment, and its implementation as a computational design tool designed to be used by architects. The tool is implemented in C# for Grasshopper (GH), a plugin of Rhinoceros CAD software. This tool minimizes the internal transportation processes between interrelated facilities where each facility is assigned to a location in an existing building. In our model, the problem of assignment is relaxed in that a single facility may be allowed to be allocated within multiple voxel locations, thus alleviating the complexity of the unequal area assignment problem. The QAP formulation takes into account both the flows between facilities and distances between locations. The distance matrix is obtained from the spatial network of the building by using graph traversal techniques. The developed tool also calculates spatial geodesic distances (walkable, easiest, and/or shortest paths for pedestrians) inside the building. The QAP is solved by a heuristic optimization algorithm, called Iterated Local Search. Using one exemplary real test case, we demonstrate the potential of this method in the context of hospital layout design/re-design tasks in 3D. Finally, we discuss the results and possible further developments concerning a generic computational space planning framework.

1. Introduction

The term Space Planning refers to the processes aimed at arranging a spatial configuration having logistics-related objectives, ergonomics, and intended user-experience of a building [1]. The spatial configuration of a building is an abstract representation of the particular ways in which the spaces inside a building are related to one another. Mathematically, spatial configurations are represented as graphs, which are typically labelled but may or may not be assumed as directed and weighted. Conventionally, if the graph in question is weighted and directed, it is called a network.

1.1. Problem definition

Hospitals consist of a wide range of functional units, each serving for different activities such as clinical, nursing, administration, service (food, laundry, etc.), research, and teaching. There are also numerous types of user groups and materials, moving or being transported between those various functions within the hospital. The flowrates of these movements in between the functional spaces can be modelled as a graph/matrix indexed by the indices of the functional units, we can consider this as a graph describing the functional requirements. On the other hand, in an existing spatial configuration, how the spaces are interrelated can be modelled as a dense graph/matrix encoding the distances

* Corresponding author. Department of Architectural Engineering and Technology, Chair of Design Informatics, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, the Netherlands.
E-mail addresses: c.cubukcuoglu@tudelft.nl, cemre.cubukcuoglu@yasar.edu.tr (C. Cubukcuoglu), p.nourian@tudelft.nl (P. Nourian), fatih.tasgetiren@yasar.edu.tr (M.F. Tasgetiren), i.s.sariyildiz@tudelft.nl (I.S. Sariyildiz), s.azadi-1@tudelft.nl (S. Azadi).

https://doi.org/10.1016/j.jobe.2021.102952
Received 17 February 2021; Received in revised form 14 June 2021; Accepted 3 July 2021
Available online 13 July 2021

between spaces of interest, computed with respect to the [temporal] length of geodesics or optimal paths; we may refer to this graph as the spatial configuration graph. Therefore, one of the challenges of space planning can be seen as matching these two graphs such that the distance between pairs is small when the flow rate between them is large and vice versa.

According to the literature, around 67% of health and care employees are not able to perform their jobs efficiently due to the unsuitable layout of the working spaces [2]. Especially nurses sometimes spend more time walking than the activities related to patient care in a day [3] because of the spatial connectivity problems of the interrelated spaces. In one study, it has been observed that 28.9% of nursing staff time wasted on walking [4]. Previous studies also show that layout type of nursing unit has an impact on the walking time of nursing staff, e.g. Ref. [5] established that nursing staff in the radial unit walked 4.7 steps per minute while the other staff working in rectangular unit walked 7.9 steps per minute, which is significantly more. The poor placement of the clinics combined with the increasingly overwhelming volume of traffic between them was causing delays and heavy congestion in hospitals [6]. All these examples show that spatial configuration has a great impact on the efficient functioning of hospital buildings in terms of walking distances and transportation processes. Therefore, providing efficient transportation processes by minimizing the walking distances between interrelated spaces should be the major concern in hospital layout planning.

The use of Quadratic Assignment Problem (QAP) is highly desirable to deal with layout planning problems in hospitals. It is a well-known problem in the field of Operations Research. Most of the facility layout problems (FLP) are formulated as QAP to minimize the transportation cost. The requirements for design in the production industry are parallel to those in hospital design. Therefore, facility-planning methodologies, which are widely used in industrial engineering, are needed in dealing with hospital layouts.

1.2. Related literature

QAP is developed by Koopmans and Beckmann (1957) [7]. It is one of the most difficult computational problems in the NP-hard class. Hence solving them optimally in a reasonable time is a very challenging task. Although exact problem-solving might be intractable, there exist heuristics algorithms capable of solving the QAP even in large sizes with nearly optimal solutions in a reasonable time.

1.2.1. QAP in general

Several solution techniques for the QAP have been suggested in the literature. As early works of QAP for FLPs, Kaku et al. (1988) [8] proposed a heuristic approach (exchange-improvement routine) for multi-story layout design. Kaku (1992) [9] proposed a procedure that combined a constructive heuristic and exchange improvement for loop conveyor and linear-track layout cases. Rosenblatt (1992) [10] developed a hybrid method that combined branch and bound framework with heuristics for equal-sized departmental layout. A novel study by Li and Smith (1995) [11] proposed a sample test-pairwise exchange heuristic procedure (STEP) for dynamic facility layout. Urban (1998) [12] presented two heuristics (multi-greedy algorithm and GRASP) for dynamic facility layout design. Ulutas and Sarac (2006) [12] addressed QAP with relocation cost thus handling a dynamic facility layout problem by developing a heuristic algorithm, called modified sub-gradient (MSG). Ramkumar et al. (2009) [13] proposed a new heuristic (iterated fast local search) for equal area layout.

Moreover, Huntley and Brown (1991) [14] combined a genetic algorithm (GA) and simulated annealing (SA) for an equal-area layout. Yip and Pao (1994) [15] proposed a hybrid technique that combined GA and SA for equal-area layout design. Bland and Dawson (1994) [16] addressed QAP for large-scale layout using a hybrid heuristic algorithm that combined SA and TS. Chiang and Chiang (1998) [17] proposed TS and SA for facility layout. Kochar et al. (1998) [18] proposed a meta-heuristic using a genetic algorithm, called HOPE, for unequal area single and multi-row layouts. Chiang (2001) [19] addressed a modified version of QAP with binary variables by proposing a TS algorithm for interdepartmental layout. Solimanpur et al. (2004) [20] used ant colony optimization for inter-cell layout. Nourelfalth et al. (2007) [21] used metaheuristic (Ant Colony Optimization with EGD local search) for equal-area layout. Jaramillo and Kendall (2010) [22] proposed a TS heuristic using different construction algorithms for machine layout. Moslemipour and Lee (2012) [23] developed a SA approach for dynamic layout. Pourvaziri and Pierreval (2017) [6] presented SA algorithm for dynamic facility layout design based on a QAP formulation. For an extensive review of solution techniques for the QAP, refer to recent survey paper by Singh and Sharma (2010) [24].

1.2.2. QAP in hospital layout planning

A relatively limited body of research has been published with respect to layout planning in hospitals.

Elshafei (1977) [25] firstly proposed QAP for locating the clinics within a hospital department using an improvement heuristic in order to optimize traveling distances of patients and delay in patient flows. Murtagh et al. (1982) [26] used QAP formulation for assigning 19 clinics to predefined locations in order to minimize transport costs by developing a new heuristic. Butler et al. (1992) [27] formulated a QAP for bed allocation in a general-purpose hospital to minimize the distance between services taken by nurses using a constructive heuristic (CRAFT). Hahn et al. (2001) [28] proposed QAP for assigning the facilities into locations in order to minimize travel distance taken by all pairs in a hospital comparing the results of several heuristics and metaheuristics solution methods. Yeh (2006) [29] focused on adjacency of objects, the distance between objects, availability of space for object location, positions of objects in relation to others for a case study of a hospital with 28 facilities. Facility layout design formulated as QAP and solved by simulated annealing with an annealed neural network. Chraibi et al. (2015) [30] minimized total traveling cost and rearrangement cost in a dynamic facility layout problem of the Operating Theatre (OT) department of a hospital. Recently, Helber et al. (2016) [31] have proposed a hierarchical modelling approach. The first stage is formulated as QAP for assigning elements to locations using a fix-optimize heuristic by considering transportation processes, locating some units on specific locations and ensuring the direct adjacency of two specific units. The second stage is detailed positioning within a location considering space requirements in a large hospital facility. Zuo et al. (2019) [32] proposed a QAP formulation for an emergency department of a hospital using a multi-objective tabu search algorithm by focusing on a real-case study.

As a result, most works focus on individual departments' 2D layout optimization in the hospital such as OT planning and nursing units. Considering the whole set of hospital departments in a 3D layout optimization is scarce in the literature. There is only one study in Ref. [31] that focuses on whole hospital departments with a practical case study. Most of the application papers considered rectilinear distances (a.k.a. Manhattan distances), not geodesic distances. To the best of our knowledge, none of the works considered the combination of the QAP with the graph-theoretical aspect of computing spatial geodesic distances (walkable, easiest, and/or shortest paths for pedestrians) inside the building in real-case scenarios.

1.3. Utilization of QAP in computational design

QAP is first developed for facility layout planning but it is also useful for spatial layout planning in architecture. There are different facility layout programs like CRAFT [33], COFAD [34], CORELAP [35], and BLOCPLAN [36]. However, these tools have rarely been utilized in architectural design practice. Considering spatial layout planning as a fundamental aspect of architectural design that affects the functional performance of hospital buildings, we argue that a systematic approach for ensuring the effectiveness and efficiency of a layout schema should be an integral part of any architectural design process, particularly in the case of critically complex building such as hospitals. The idea of bridging the gap between both the parametric CAD platforms and the layout design is not entirely new. There is a limited number of tools available in Parametric CAD platforms to facilitate the space planning processes, namely DeCoding Spaces, SpiderWeb, and Syntactic [37–39]. To the best of our knowledge, none of the tools proposed the QAP approach in layout planning and none of them utilize the shortest paths for pedestrians. These three toolkits provide methods for formulating space planning problems and analysing spatial configurations within the framework of Space Syntax theories. However, other than heuristic force-directed graph drawing solvers, none of these tool suites proffer an explicit formulation of the space planning problem as an optimization problem. Our proposed methodology formulates the problem of layout optimization as a Quadratic Assignment Problem, proposes a measure of quality as the Logistic Cost Function, as to which a benchmark can be created for the improvements on the inner walking/transportation costs within a complex layout. Finally, the proposed solver reduces these costs to a minimum and finds a new spatial configuration.

1.4. Contributions

The main goal of this paper is to formulate and solve a 3D space planning problem methodology in the form of a Quadratic Assignment Problem (QAP), in the context of a re-design/renovation task, by considering the effect of geodesic distances through a network of circulation spaces in 3D. The QAP is a well-known problem in the field of Operations Research from the category of the facility's location/allocation problems. The methodology is first and foremost developed for hospital space planning; however, it can also be used in design and optimization for other types of complex buildings, especially in order to study reuse scenarios. The methodology is implemented and tested partly in C# language [40] for McNeel's Grasshopper3D [41] and partly in VEX language [42] for SideFX′ Houdini [43]. The following steps are presented in this paper:

- Formulating a layout problem in architectural design as a QAP based on geodesic distances
- Introducing a practical CAD workflow for applying QAP solvers to 3D spatial layout problems
- Discretizing and modularizing the design space as a way of structuring the geodesic computation problem in 3D as well as relaxing and simplifying the unequal area QAP problem
- Estimating flows and entering the flow matrix as an input of the tool
- Calculating spatial geodesic distances and entering the distance matrix as an input of the tool
- Proposing a heuristic algorithm for solving the QAP
- Implementation of the model and heuristic optimization algorithm
- Running the solver, reflecting on the results, and comparing them to the existing state of an actual hospital

1.5. Gaps in the literature

The detailed literature review is given in section 1.2. We have identified and summarised the following gaps in the literature concerning the layout optimization of existing hospitals:

- There is no complete methodology for a 3D layout problem with a QAP formulation, most papers are focused on 2D layout and focused on specific departments such as the operation theatre and the nursing units. However, we focus on the entirety of the hospital design problem.
- There is no study on QAP that focuses on geodesic distances (especially in 3D). Most studies focus on rectilinear distances on a flat 2D plan.
- The combination of QAP with the graph theoretical aspects (wayfinding) in real-world test-cases is unique.
- Modularization/Discretization of space layout problem both in terms of modularization of the departments and the walkable space in between the departments in 3D for solving a QAP.
- There is no implementation of QAP problem-solving in computational design methodologies.
- The Iterative Local Search algorithm is not entirely new, but it has not been implemented in the case of space layout of hospitals.

2. Problem formulation

As explained in the literature review the Quadratic Assignment Problem is well known in the field of Facilities Layout Planning but relatively unknown in architectural layout design. Interestingly, when the focus of the optimization task is on improving the configuration of an existing building, the Quadratic Assignment Problem arises naturally when we consider the efficiency of the spatial configuration of the building as a whole. QAP is a combinatorial optimization problem that aims at allocating a set of facilities to a set of locations such that the total transportation cost is minimized. Total transportation cost is a function of the flows between facilities and the distances between locations. The QAP has a discrete representation of areas since the facility departments can only be assigned to predefined network locations. Essentially, the problem of assignment here is an unequal area layout problem because each department has a different required amount of surface area Table 1 (Appendix-A). Our model makes this discretization regular by topologically abstracting the whole 3D walkable floor space of the building as a voxelated domain [44,45]. Voxels or (volumetric pixels/picture cells) are 3D regular units of space for partitioning a 3D volume into a Cartesian grid of cells. The design domain, in this case, is voxelated for two main reasons: 1) to modularize the units of space so that the problem of layout can be formulated as an assignment problem, and 2) to use the explicit topological relations between adjacent voxels for constructing a network model of space to compute the geodesic distances for the computation of the logistic cost function (the objective function). In accordance with this tessellation, at a higher resolution corresponding to the structural grid of the building, we break down the departments in terms of their surface area requirements into 'multiple facilities, each of which has exactly the area of one pixel'. Such a regular tessellation brings about two main advantages: on the one hand computation of geodesic distance becomes straightforward on the network generated from the voxels, and on the other hand the problem of assignment is relaxed in that a single facility may be allowed to be allocated within multiple voxel locations, thus alleviating the complexity of the unequal area assignment problem. Each facility can be assigned to exactly one location and no location is assigned to more than one facility. Therefore, the number of facilities should be the same as the number of locations. The basic QAP model can be formulated as:

$$\underset{\boldsymbol{\pi} \in P_n}{\text{minimize}} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} D_{i,j} T_{\pi_i,\pi_j}$$

Where, $\boldsymbol{\pi}$ is a vector of integers denoting a permutation of facilities at a moment in time, T_{π_i,π_j} denotes the [transportation] flow between a permuted facility $\pi_i = k$, and another permuted facility $\pi_j = l$ and $D_{i,j}$ is the distance between location i and j. P_n denotes the set of all permutations $\boldsymbol{\pi}^{(t)} : \mathbb{N} \to \mathbb{N}$, where the superscript denotes the iteration time. Note that the cardinality of P_n will be $n!$, and so a brute-force search becomes intractable as soon as the problem gets large, i.e. an order of complexity of $\mathscr{O}(n!)$; which in the case of our example would be about searching within $64! = 1.2688693e + 89$ possible permutations. Note that the objective function is measuring the expected travelled distance for a typical building user; this is because the transition probabilities (e.g. those given as percentage values in Table 2, Appendix-B) are dimensionless/unitless and that the travelled distance between every two nodes is multiplied with the probability of that transition.

3. Methodology

We have formulated the problem as a matter of reducing the logistic cost function by choosing the right permutation of facilities within a set of existing locations. Due to the physical nature of the costs (distances) and the dimensionless (unitless) meaning of the flow rates, the physical unit of the objective function is the same as the distances. This means that the distances in between the departments in a 3D space must be computed. In order to do so, we propose the following methodology:

1. Estimate the flow-rates as the transition probability between the departments for a pedestrian (the flow-rates, in this case, are considered as given, see Appendix B);
2. Discretise the walkable space in between the departments and model the topology of the connectivity between the discrete spaces as a graph/network;
3. Compute the geodesic distance between the department locations in the discretized space and attribute the distances to the pairs of departments using the Floyd-Warshall algorithm;
4. Heuristically improve (minimize) the objective cost function by try-fitting various permutations of facilities over the locations using the Iterative Local Search algorithm.

3.1. Flows

As stated above, the objective of the QAP is a function of flows between facilities and distances between locations as constant parameters and the only variable parameter is the permutation, hence it is called a combinatorial optimization problem. The flow information refers to any quantitative relationship score between any pair of items, typically the estimated volume of transporting materials or probability of transition of medical staff in between facilities. We have considered the latter probabilistic interpretation in our formulation. This score can also be interpreted from given adjacency requirements of a hospital, which are typically recommended for ensuring effective logistics according to medical procedures, providing privacy or community, security, safety, hygiene, congruence of noise levels, etcetera. The so-called RELChart table, e.g. the one used in this paper (Appendix B) is a matrix whose entries indicate the relative importance of closeness between two departments. Considering the central importance of this matrix in the formulation of the QAP problem or even only in assessing the quality of a particular assignment, it is important to compose this table with objective information. However, in practice, such tables are often composed following discussions of the board of directors of a hospital. Nevertheless, for a building that does not exist yet, figuring out such importance ratings is a daunting task. For an existing building, however, these relative importance ratings can be replaced with the measured or estimated probabilities of transition between pairs of departments objectively. Given the technical difficulty of measuring such probabilities in practice, estimating the probabilities according to the foreseen procedures is an alternative that has been shown to be feasible using a Discrete Event Simulation (q.v. [46,47]).

The number of spatial units must remain the same throughout the QAP solving, and so, we firstly split the departments into spatial units considering the size of the designated modules and their area requirements. Then we divide the predicted flow rates mentioned in the RELChart equally between the dividend units. We added this description to the methodology section. The suggestion to set high intra-closeness ratings for encouraging closeness between the split parts is very logical and it is already implemented by setting high flow rates between the divided units (100% closeness). The numbers written in our REL chart are percentages [0,100] which are interpreted at the end as numbers in the range of [0,1] as transition probabilities. This makes it possible to have a physical interpretation of the objective cost function as an expected travelled time as explained in the section Application. However, if we raise these numbers to values higher than 100 (or 1) this would disrupt the probabilistic interpretation and make it hard to explain and justify the results.

3.2. Spatial geodesic distance

In order to compute walking distances within a building, spatial geodesic (optimal paths on a network) are used in this method. In the mathematical field of graph theory, the geodesic distance between two vertices in a graph is the total sum of the costs attributed to the edges connecting them through an optimal path. For constructing such paths and computing geodesic distances, firstly, the set of all spaces (locations, corridors, and stairs) are discretized in a surface geometric model [45]. The spatial network of the indoor walkable space of the building is then extracted from this mesh, a multi-source graph-traversal search is run, and the distance matrix is obtained. The graph is constructed based on 6-neighbourhoods of voxels (i.e. voxels connected to their top, bottom, left, right, back and front neighbours) [48]. Then multiple A* searches are run within the constructed graph to find the geodesic distance from every location to every other location. Note that the location points (as marked in red and shown in Fig. 1) are exactly 64 voxels corresponding to the larger sets of voxels whose areas are equal to 100 m^2. These large

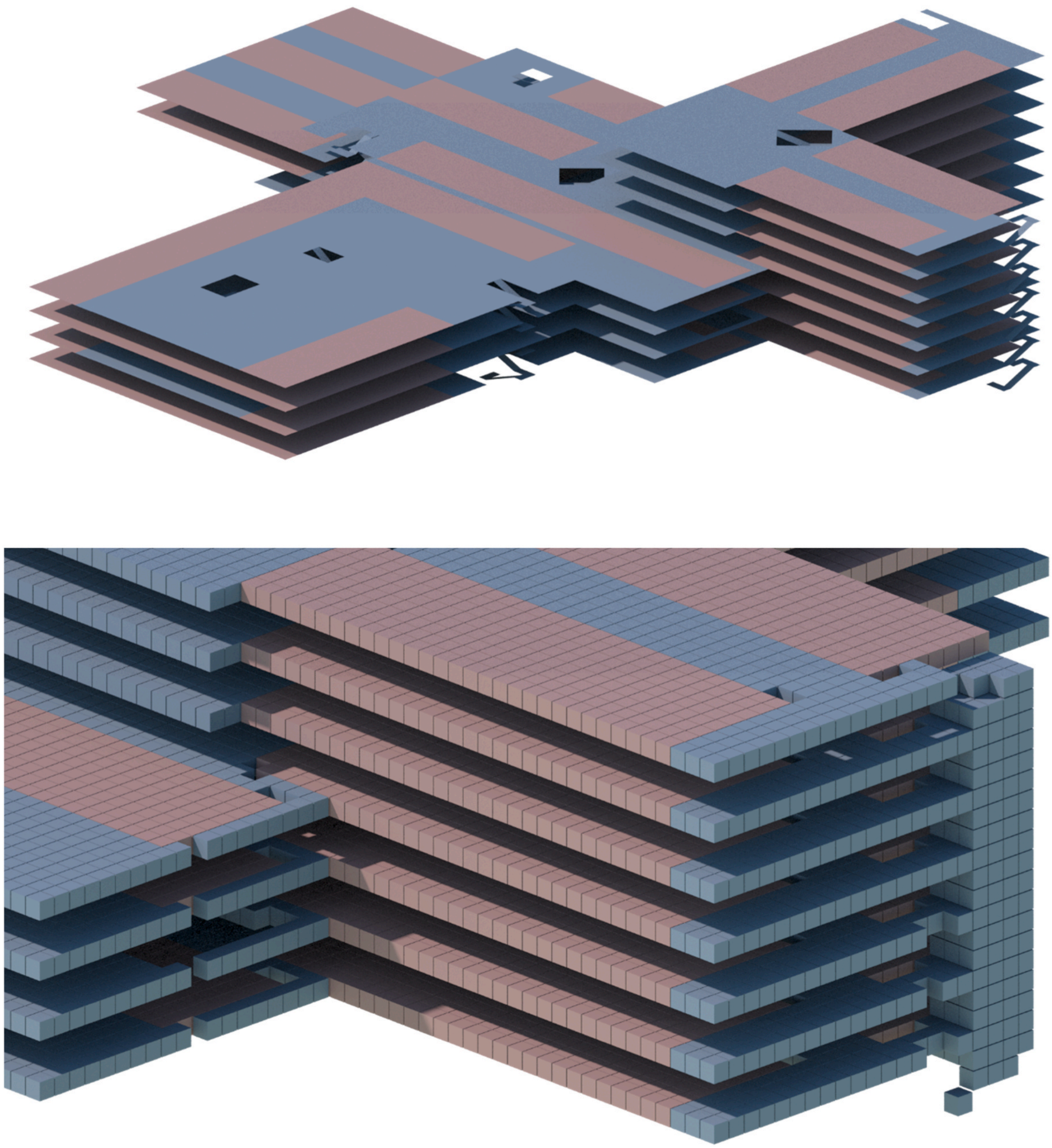

Fig. 1. The walkable space as a mesh (top) and its discretized voxel model (bottom); blue voxels: circulation areas & red voxels: location areas. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

areas are not included in the graph generated from the voxels because the model only needs to have the distance from their access points to other access points.

3.3. Heuristic problem solving

As stated before, QAP is an NP-hard problem. Therefore, heuristic optimization algorithms are seen as remedies for tackling this complex problem in large instances. In this tool, we selected Iterated Local Search algorithm (ILS) [49–52] for problem solving. Recently, the performance of an ILS algorithm in Ref. [53] has been tested on QAP instances arising from real-life problems as well as on several benchmark instances from the QAPLIB [54]. Inspired by Refs. [53,55], we utilize the ILS algorithm for the space planning tool presented in this paper. Details of the algorithm that is considered in this paper are given in sub-sections below.

3.3.1. Solution encoding

The encoding scheme in our algorithm corresponds to a sequence of integers that represents facilities in a feasible solution (permutation).

3.3.2. Initial solution

In the Iterated Local Search (ILS) algorithm, the initial solution $\boldsymbol{\pi}^{(0)} = [\pi_1, \pi_2, .., \pi_n]^T$ is constructed randomly, which is a permutation of the integers between 1 and n, where n is the number of facilities.

3.3.3. Perturbation scheme

In the ILS procedure, the initial solution is perturbed with swap and insertion neighbourhoods to escape from local minima. In this paper, random swap and insertion neighbourhoods are employed. The swap operator exchanges two facilities in a solution, whereas the insertion operator removes a single facility from a solution and inserts it into a random position in the solution. As an example, to a swap operator, suppose that we are given a current solution $\boldsymbol{\pi}^{(0)} = [5, 4, 2, 1, 3]^T$. Two facilities are randomly selected and they are exchanged. As an example, we randomly choose the facility $\pi_4^{(0)} = 1$ and $\pi_2^{(0)} = 4$ in order to swap them. Thus, we end up with a solution as $\boldsymbol{\pi}^{(1)} = [5, 1, 2, 4, 3]^T$. In addition, as an example of an insertion operator, we apply forward or backward insertion with an equal probability. Suppose that we are given a current solution $\boldsymbol{\pi}^{(0)} = [5, 4, 2, 1, 3]^T$. Assume that we randomly choose $\pi_3^{(0)} = 2$. Then, we remove it from the solution and insert it into the fourth position as a forward insertion to generate a new solution $\boldsymbol{\pi}^{(1)} = [5, 4, 1, 2, 3]^T$ whereas in the backward insertion, we remove $\pi_3^{(0)} = 2$ from the current solution and insert into the second position as $\boldsymbol{\pi}^{(1)} = [5, 2, 4, 1, 3]^T$.

3.3.4. Local search

After the "perturbation" of the current solution, we apply a "local search" based on swap neighbourhood. In the swap local search, the perturbed solution $\boldsymbol{\pi}^{(1)}$ goes under a swap local search procedure. The iteration counter is fixed at 1 at the beginning, we select two facilities randomly and simply swap them. If the new solution obtained after the swap neighbourhood is better than the current solution, it is replaced with the current solution and the iteration counter is again fixed at 1, otherwise, we keep the current solution as it is. And the iteration counter is increased by 1. The swap local search is repeated until the iteration counter is reached at the number of facilities n. The pseudo-code of the swap local search is given in Algorithm 1.

Algorithm 1. Swap Local Search

Input Notation	*[Data-Structure] Data Type*	*Input Name: Notes*
$[D_{i,j}]_{n\times n}$	*Matrix of float*	*Distance Matrix: a matrix whose entries represent the network distance between locations i & j*
$[T_{k,l}]_{n\times n}$	*Matrix of float*	*Flow Matrix: a matrix whose entries represent the [transportation] flows between facilities k & l*
$\boldsymbol{\pi}$	*Array of Integer*	*Current Permutation: labelling of locations as to facility indices/labels, passed by reference*
Output Notation	***[Data-Structure] Data Type***	***Output Name: Notes***
$\boldsymbol{\pi}$	*Array of Integer*	*Next Permutation: labelling of locations as to facility indices/labels*
$f(\boldsymbol{\pi})$	*float*	*Best fitness: the best fitness value found by the algorithm*

Problem: Given a distance matrix $\mathbf{D}$ *on n locations and a transportation flow matrix* $\mathbf{T}$ *on n facilities, desired a permutation* $\boldsymbol{\pi}$ *of locations such that the fitness* $f(\boldsymbol{\pi}) = \sum_{i=1}^{n}\sum_{j=1}^{n} D_{i,j} T_{\pi_i,\pi_j}$ *is minimized.*

```
Procedure SwapLocalSearch(π, f(π))
  iteration = 1
  while(iteration ≤ n)do
     Find two facilities, π_a and π_b, randomly
     Swap π_a and π_b
     Generate a new solution as π̂ with swapped π_a and π_b
     if(f(π̂) < f(π)) then do
          π = π̂
          iteration = 1
    else
       iteration = iteration + 1
    endif
  endwhile
  return (π, f(π))
end procedure
```

Input Notation	*[Data-Structure] Data Type*	*Input Name: Notes*
$[D_{i,j}]_{n\times n}$	*Matrix of float*	*Distance Matrix: a matrix whose entries represent the network distance between locations i & j*
$[T_{k,l}]_{n\times n}$	*Matrix of float*	*Flow Matrix: a matrix whose entries represent the [transportation] flows between facilities k & l*
$\boldsymbol{\pi}$	*Array of Integer*	*Current Permutation: labelling of locations as to facility indices/labels, passed by reference*
Output Notation	***[Data-Structure] Data Type***	***Output Name: Notes***
$\boldsymbol{\pi}$	*Array of Integer*	*Next Permutation: labelling of locations as to facility indices/labels*
$f(\boldsymbol{\pi})$	*float*	*Best fitness: the best fitness value found by the algorithm*

Problem: Given a distance matrix $\mathbf{D}$ *on n locations and a transportation flow matrix* $\mathbf{T}$ *on n facilities, desired a permutation* $\boldsymbol{\pi}$ *of locations such that the fitness* $f(\boldsymbol{\pi}) = \sum_{i=1}^{n}\sum_{j=1}^{n} D_{i,j}T_{\pi_i,\pi_j}$ *is minimized.*

$\textbf{Procedure}\ SwapLocalSearch(\boldsymbol{\pi}, f(\boldsymbol{\pi}))$
  $iteration = 1$
  $\textbf{while}(iteration \le n)\textbf{do}$
    $Find\ two\ facilities, \pi_a\ and\ \pi_b, randomly$
    $Swap\ \pi_a\ and\ \pi_b$
    $Generate\ a\ new\ solution\ as\ \hat{\pi}\ with\ swapped\ \pi_a\ and\ \pi_b$
    $\textbf{if}(f(\hat{\boldsymbol{\pi}}) < f(\boldsymbol{\pi}))\ then\ \textbf{do}$
      $\boldsymbol{\pi} = \hat{\boldsymbol{\pi}}$
      $iteration = 1$
    $\textbf{else}$
      $iteration = iteration + 1$
    $\textbf{endif}$
  $\textbf{endwhile}$
  $\textbf{return}\ (\boldsymbol{\pi}, f(\boldsymbol{\pi}))$
$\textbf{end procedure}$

The general framework of the ILS algorithm is given in Algorithm 2. Briefly, the initial solution is constructed randomly. Then, a swap local search is applied to the initial solution. A loop-based on the termination criterion is started. Repeatedly, perturbation and swap local searches are applied to the current solution until a termination criterion is satisfied. If we need to implement a fixed-department constraint, we can define a set dubbed f and we can add a condition to the Swap procedure and change it to:

$if(\ a \notin f \wedge\ b \notin f)\ then\ Swap\ \pi_a\ and\ \pi_b$

Algorithm 2. The General Framework of the Iterated Local Search Algorithm

Input Notation	[Data-Structure] Data Type	Input Name: Notes
$[D_{i,j}]_{n\times n}$	*Matrix of float*	*Distance Matrix: a matrix whose entries represent the network distance between locations i & j*
$[T_{k,l}]_{n\times n}$	*Matrix of float*	*Flow Matrix: a matrix whose entries represent the [transportation] flows between facilities k & l*
$\boldsymbol{\pi}^{(0)}$	*Array of Integer*	*Initial Permutation: labelling of locations as to facility indices/labels*
Output Notation	**[Data-Structure] Data Type**	**Output Name: Notes**
$\boldsymbol{\pi}^{(t)}$	*Array of Integer*	*Last Permutation: labelling of locations as to facility indices/labels*
f	*float*	*Best fitness: the best fitness value found by the algorithm:* $f = \sum_{i=1}^{n}\sum_{j=1}^{n} D_{i,j}T_{\pi_i,\pi_j}$

Problem: Given a distance matrix $\mathbf{D}$ *on n locations and a transportation flow matrix* $\mathbf{T}$ *on n facilities, desired a permutation* $\boldsymbol{\pi}$ *of locations such that the fitness* $f(\boldsymbol{\pi}) = \sum_{i=1}^{n}\sum_{j=1}^{n} D_{i,j}T_{\pi_i,\pi_j}$ *is minimized.*

$$
\begin{array}{l}
\textbf{Procedure}\ Iterated\ Local\ Search(\boldsymbol{\pi}^{(0)}, f(\boldsymbol{\pi}^{(0)}), t\_max)\\
\quad t = 0\\
\quad \boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(0)}\ \#GenerateInitialSolution\\
\quad \textbf{Repeat}\\
\qquad \overline{\boldsymbol{\pi}} = Perturbation(\boldsymbol{\pi}^{(t)})\\
\qquad \boldsymbol{\pi}^{(t)} = SwapLocalSearch(\overline{\boldsymbol{\pi}}, f(\overline{\boldsymbol{\pi}}))\\
\qquad t = t + 1\\
\quad \textbf{Until}(t < t\_max)\\
\quad \textbf{return}\ (\boldsymbol{\pi}^{(t)}, f(\boldsymbol{\pi}^{(t)}))\\
\textbf{End}\ Iterated\ Local\ Search
\end{array}
$$

4. Test & implementation

The purpose of our implementation at this point was to test the algorithms. More specifically, the purpose was to verify whether the algorithms work as expected in terms of the correctness of the results and to validate whether the results are improved. We have implemented the method presented above partly using the C# programming language and developed a space planning tool, called QAP Solver as an add-on for McNeel's Grasshopper3D [41] software application and partly as a VEX add-on for SideFX′ Houdini [43] for computing network geodesics and the corresponding distance matrix.

QAP Solver component implements Algorithm 2 for solving a QAP instance based on the given input data (flow and distance matrices). Inside the component, the first initial solution is generated randomly, and then the optimization algorithm is run after Boolean toggle is set to "True" mode; this will trigger the generation of new permutations, for each of which the objective function of the QAP is evaluated and reported. In each generation, the component is capable of showing the change of decision variables (permutation) on a collection of number sliders by realizing a slider update procedure inside the component. In this way, users of the tool can see how well the layout is being improved over the generations in terms of the value of the fitness function and at the same time see the generations in real-time, as the number sliders that encode the permutation are used to pick and change the colours of rooms in the 3D model. In addition, the tool allows users to set the maximum number of trial times (t_max in Algorithm 2). The user is expected to connect as many number sliders as the number of facilities (functional units). All sliders should be connected to the QAP Solver component. The output of the component presents the result of the optimization as well as the amount of improvement in objective value. The permutation results are also shown on the number sliders, e.g. if the first slider has resulted as 2, then the second facility is placed to the first location and so on. The QAP Solver component is shown in Fig. 2 when working on a toy problem with three facilities to be assigned to three locations.

For computing the geodesic distances, we first extract a set of meshes representing the connective spaces such as corridors, stairs, and ramps [if any]; then we voxelate these spaces using openVDB [56]; then construct a network out of the voxels, and then calculate shortest paths from all locations to each other location, i.e. the same set of locations will be used as both origins and as destinations. Technically, the locations are first mapped onto their closest points/voxels on the network. The output of the process will be the matrix of distances $[D_{i,j}]_{n\times n}$ where n is the number of locations. This output is directly used in the QAP solver. The working principle of the QAP solver can be shown in the video available online (https://www.youtube.com/watch?v=Lv52qy1OjSw).

5. Application

In this section, we articulate an outlook for using the QAP tool in space planning and design of existing hospitals in a larger context.

Due to the nature of the QAP method, it can be used in case of the following scenarios in redesigning a hospital building, i.e. a building with a set of facilities is to be moved to another or the same building with the same number of locations:

1. When we know the transportation flows [logistics of pedestrians or materials] between the facilities of a building;
2. When we can estimate the flows between the facilities of a building, e.g. by utilizing a simulation procedure such as Discrete Event Simulation [46].

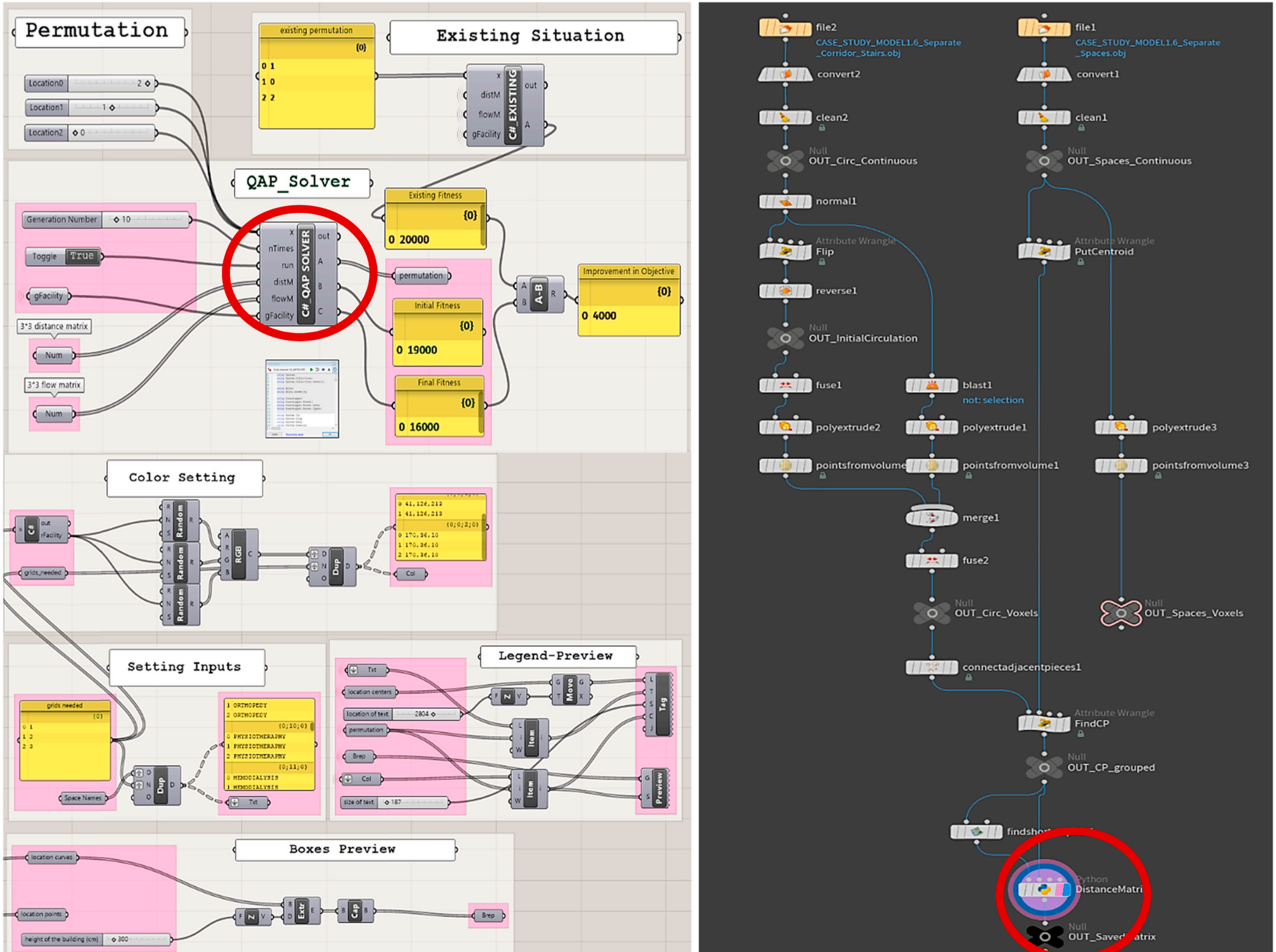

Fig. 2. QAP_Solver component in Grasshopper3D (left) and the geodesic/network distance computing in houdini (right).

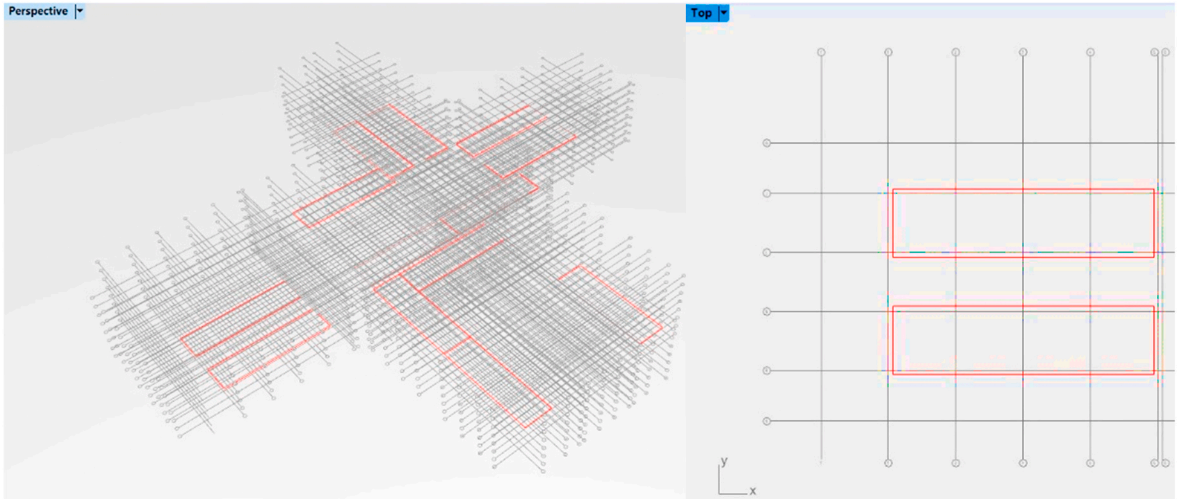

Fig. 3. Definition of rectangular location boxes' floor surfaces (consisting of 4 structural grid pixels based on building axes).

We illustrate the application of the proposed QAP method can be used in computational space planning in the context of an exemplary case study hospital (corresponding to the first scenario described above). All hospital departments are considered in the layout optimization problem; however, it would be also possible to perform a similar procedure on the inner spatial layout of only one department like Operation Theatre layout or Intensive Care Unit layout, i. e QAP at a higher level of detail. The chosen hospital is a state hospital with a capacity of 250-beds for the in-patient wards. It is a 9-story building including basement and ground floors. In this model, we excluded the locations placed on the basement and the 7th level of the building in the existing situation; because, the 7th level has only a terrace area, which is not suitable for placing any facility; and that the spaces located in the basement have some specific features and their locations cannot be changed. These spaces that are excluded from the model can be listed as a mortuary, a worship-space, a bunker, parking lots, and storage. Furthermore, some of the functional units may require some specific locations due to a specific feature in hospitals, e.g. the emergency department and main entrance should be at the ground level of the hospital. The locations of these facilities are excluded in the model by making their locations constant in proper locations. Based on this, there are 34 facilities considered in this case model for renovating the hospital layout.

Utilizing the QAP method is limited, theoretically, to equal-area layout problems. However, each functional unit differs with respect to its space requirements in this case. Although this is arguably an inherent limitation of the methodology, we can relax the requirements such that this is no longer a limitation. Since each location is represented with [modular] discrete spaces (boxes in 3D), the number of needed boxes for each facility can also differ. As a new approach to adapt QAP to unequal departments, we can repeat each facility according to the area requirement in the flow matrix and distribute/divide the flows accordingly. For instance, assuming that each box has a capacity of 100 m^2 and that the cardiology department needs 300 m^2 then we can define this department 3 times in the flow matrix and then divide the row and the column corresponding to this space in the flow matrix by 3 and use the results in the new rows and columns. By this repetition, the number of facilities becomes the same as the number of locations, which is defined as 64 in the model; while this also entails that a single facility may not necessarily stay at a single location. This is obviously a limitation but at the same time, it might be beneficial in light of a higher level of satisfaction with the logistics requirements. Based on the structural system of the building, we define the locations as rectangular spaces, in line with the structural axes. This ensures that during the renovation the structural system does not have to be modified. Afterward, we tessellate each rectilinear floor space into four quadrangular faces. Each quad face refers to a square-like space surrounded by vertical columns axes and horizontal beam axes of the existing building (as shown in Fig. 3). A list of spaces with the number of needed rectangular boxes is given in Appendix-A. The flow matrix is given in Appendix-B. For calculating the distance matrix, the spatial network of the building is given in Fig. 4.

The RELChart table in Appendix-B is a matrix whose entries indicate the relative importance of closeness between two departments. Considering the central importance of this matrix in the formulation of the QAP problem or even only in assessing the quality of a particular assignment, it is important to compose this table with objective information. However, in practice, such tables are often composed following discussions of the board of directors of a hospital. Nevertheless, for a building that does not exist yet, figuring out such importance ratings is a daunting task. For an existing building, however, these relative importance ratings can be replaced with the measured or estimated probabilities of transition between pairs of departments objectively. Given the technical difficulty of measuring such probabilities in practice, estimating the probabilities according to the foreseen procedures is an alternative that has been shown to be feasible using a Discrete Event Simulation in Ref. [47]. The particular table added in the appendices of this paper, however, is a RELChart produced by collating expert interviews, site visits, design guidelines/standards, and recommendations from the scientific literature.

Regarding the computational results, the proposed heuristic algorithm for the QAP is tested on an Intel Core-i7 computer, with 2 GB of RAM. Maximum trial time is taken as 50 000 iterations with a seed number 5. Permutations of the existing and proposed layout are given detailly in Appendix-C. Based on this table, existing fitness is 483751000 expected travelled steps (roughly equal to 60 cm, i.e. the small voxel-size in the model). After the optimization by QAP Solver, the new fitness is 411578400. The improvement in the objective value is 72172600 steps

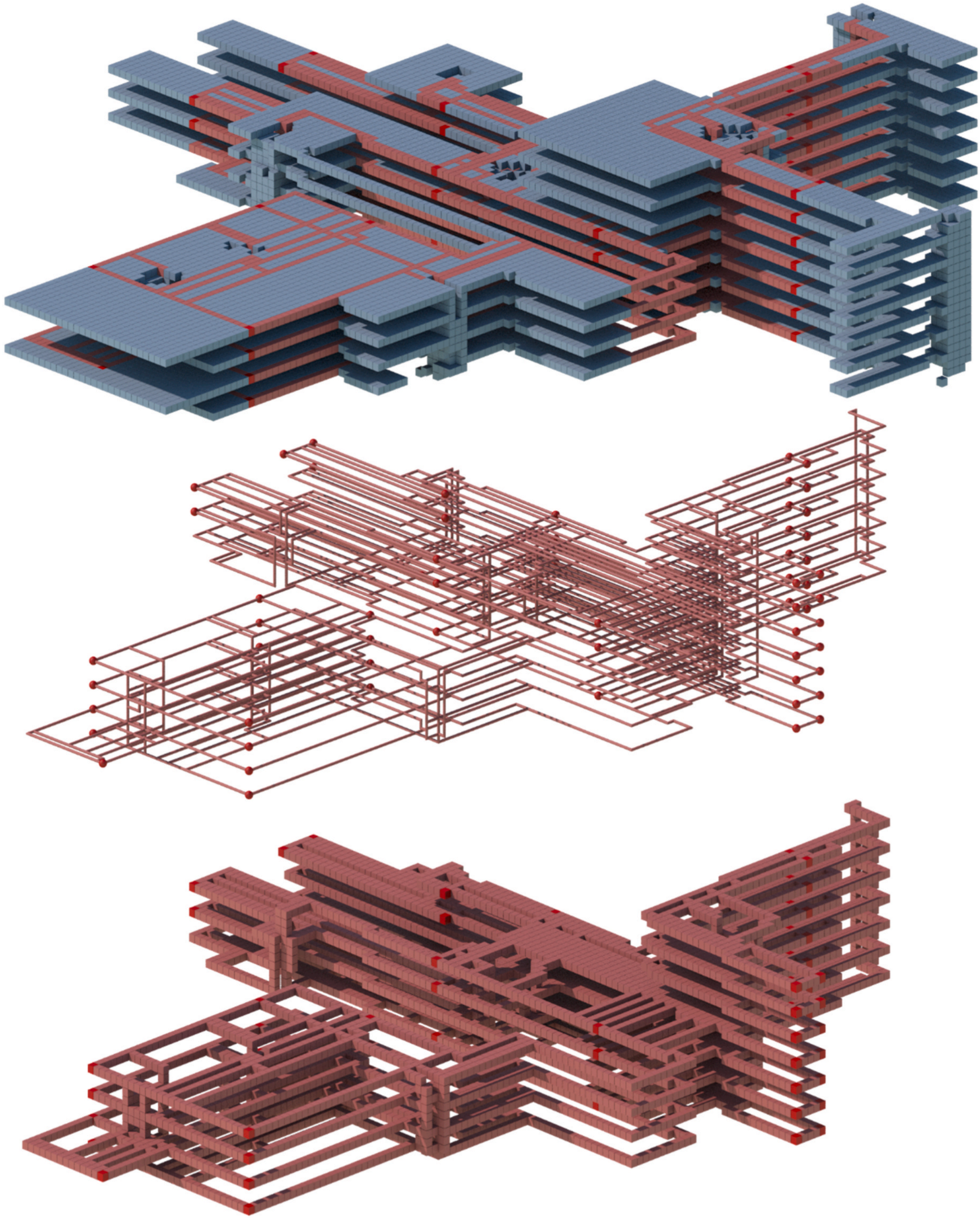

Fig. 4. Extracting "spatial network" of the building (top), continuous version of paths (middle), a discrete version of paths (bottom), (red dots are location points). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(adjusted to 721726.00 after dividing by 100 for converting flow percentages to probability fractions), which equals about 469121.90 m of travelled distance. To put this result in a more concrete context, let us assume that an average person can walk 5 km per hour; then this number means that we have reduced the time spent for walking in between the facilities by 469121.90/5000 = 93.82438 person hours for a typical day. Note that as we explained in the definition of the unit of the objective function, this is the 'expected travelled distance' (or the time spent on walking) for a typical building user on a typical day of operation. This means that the improvement can be attributed to the building as a whole

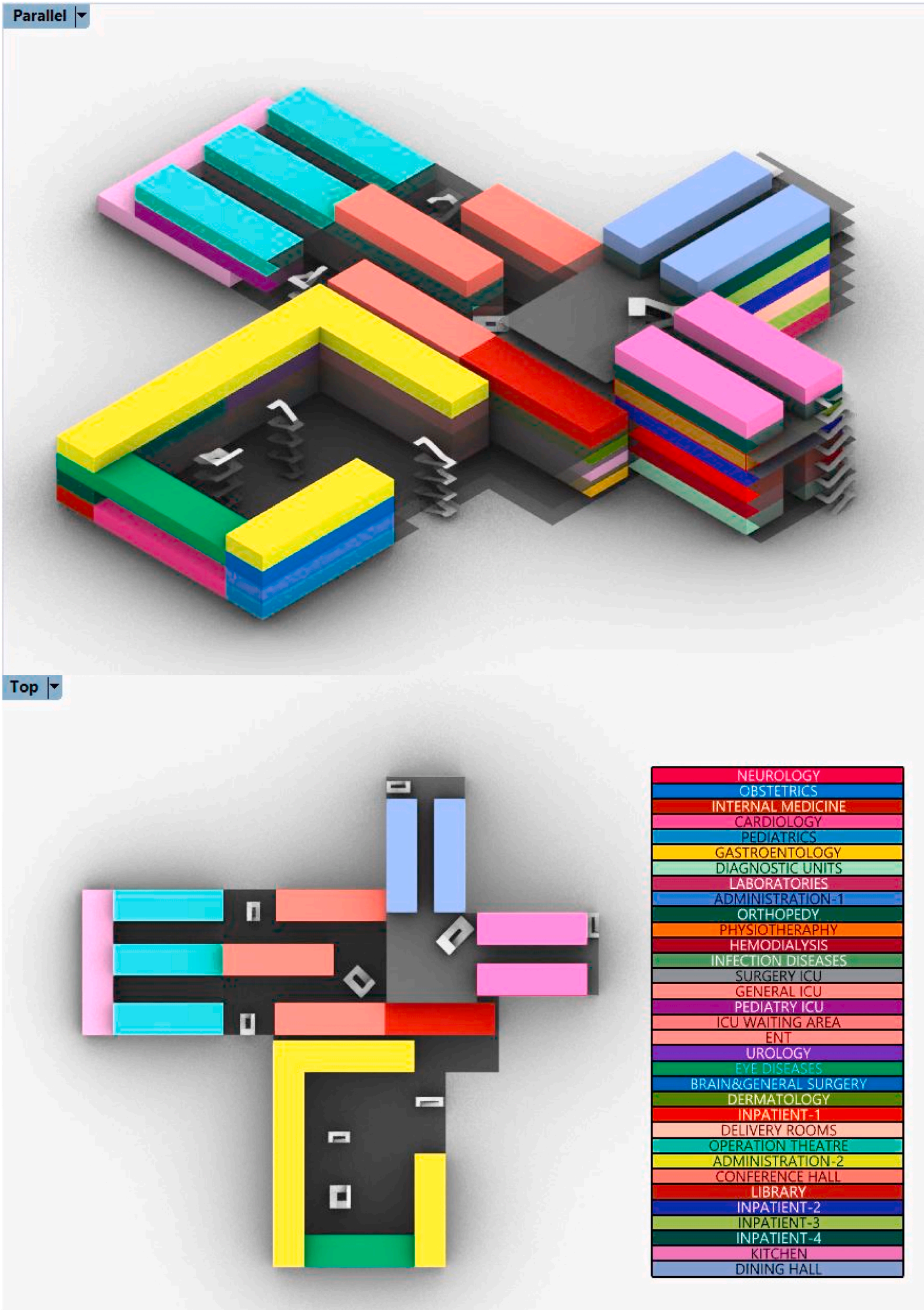

Fig. 5. Existing layout.

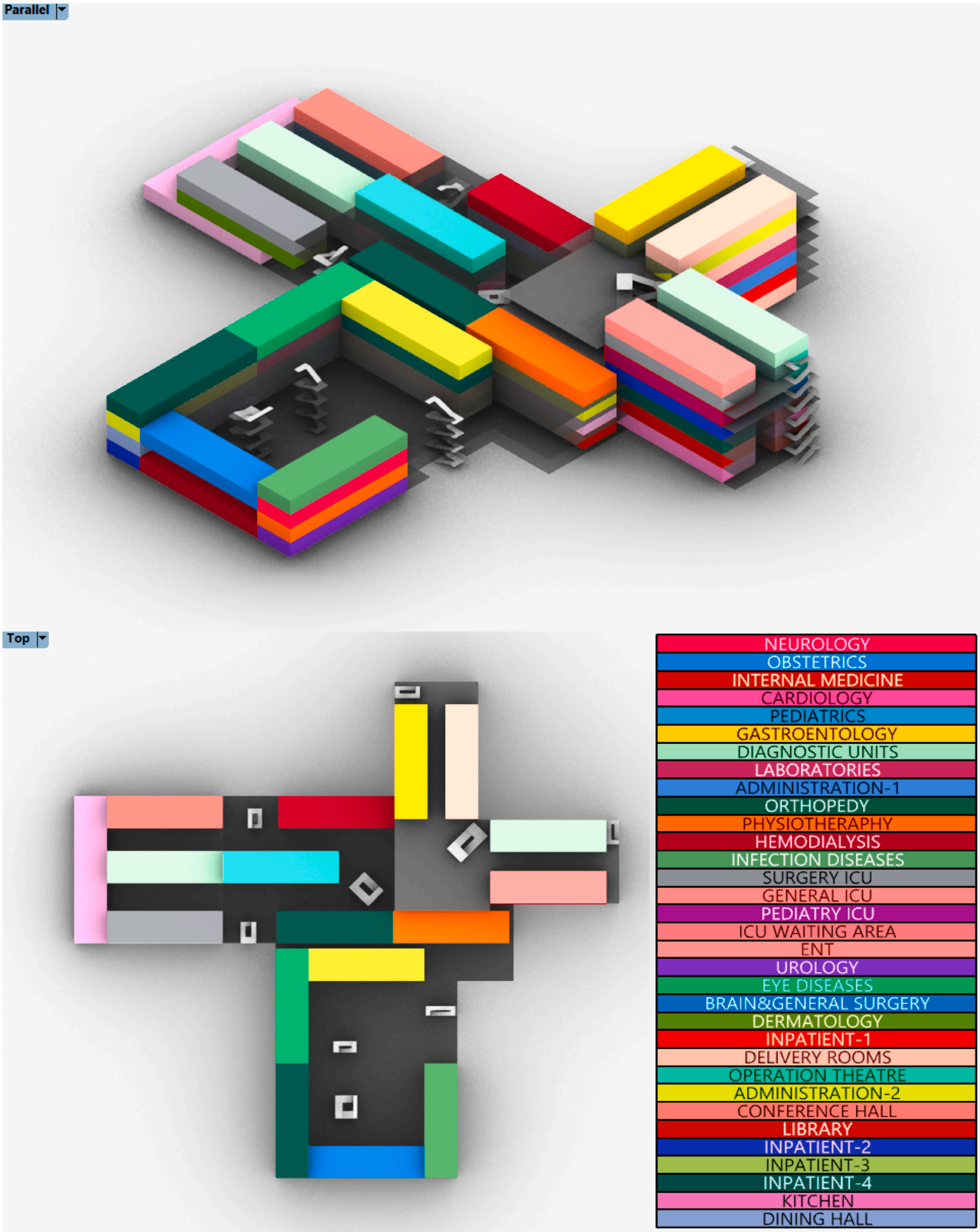

Fig. 6. Proposed layout.

rather than an individual person. The existing layout and the new layout result for the case study hospital model are visualized in Figs. 5 and 6.

Regarding the design results, the placement of the inpatient areas is still located at the top levels of the building as expected due to the daylight requirements of patient wards. However, in the proposed layout, interrelated spaces with inpatient have moved to various locations. In the proposed layout, operation rooms are located closer to the patient wards and there exists ease of access between these facilities with a vertical short connection. Critical patient flow processes and the shared equipment entail better access between intensive care units and operating rooms. In the new layout, intensive care units and operation rooms have a better connection since Intensive Care Units are the neighbour of the Operating Theatre at the upper levels of the building. In addition, delivery rooms have to get closer to Operation Theatre, which is an advantage for shared staff and facilities. The kitchen and dining hall were located at the last level of the building whereas they are located at lower levels in the proposed layout. This provides quicker food transportation from the kitchen to the medical spaces e.g. dining & kitchen area has horizontal access to outpatient and vertical access to inpatient. In the new layout, all departments related to maternity like Paediatrics Intensive Care Unit, Paediatrics Outpatient, Delivery Rooms and Obstetrics departments are located close to each other.

Laboratories and diagnostic units are located adjacent to each other on the ground floor and become closer to the outpatient departments as expected. In addition, inpatient departments were placed in the areas that have one single corridor in the old layout. Whereas, these departments are mostly located in areas with a radial layout structure. Surgical outpatients like Obstetrics, Orthopedy, and Urology departments are located closer to the operating rooms in the new layout. Transportation processes between interrelated spaces became more efficient with the proposed approach.

6. Conclusions

This paper introduces a new computational space planning methodology based on the well known Quadratic Assignment Problem, presents a heuristic solver for it and presents the test results on a hospital redesign case study. One of the novelties of the presented methodology is that it utilizes the spatial network of the existing building for computing geodesic distances, which are then used as inputs of the QAP model. Results show that objective value is reasonably minimized, and design results seem more logistically efficient. We have estimated an aspect of the operational cost of the building as expected travel time of employees/users of the building using our objective cost function. It must be noted that this is a matter of ex-ante assessment and not a measurement, as measuring the actual travel time would require tracking the personnel inside the building and fall out of the scope of this paper. The time saving achieved with our methodology based on Operations Research has achieved an estimated reduction of around 90 person-hours for a typical operational day of the hospital. Due to its aggregate nature, such a reduction should be of interest for the management of the hospital as it implies not only a reduction of costs but also implicitly an increased comfort for the employees, users, and thus a higher-quality service. The contributions of the paper can be recapitulated as below:

- The obtained results, i.e. the new configuration and its corresponding logistic cost function, reveal a major difference made by reassigning the departments to alternative locations, hence validating the major contribution of the paper on improving existing layouts. The reduction of the logistic cost function in this case corresponds to a total reduction of around 93 person-hours of expected travel time between departments of the hospital for a typical day.
- We have considered the physical constraints pertaining to the size of the departments and assumed that the departments can be accommodated into modular/rectilinear spatial units (colored in the pictures). The newly found assignment can be visually inspected from the point of view of an architect/manager and it seems to be a feasible/logical assignment in terms of other constraints that are not taken into account in this formulation. If the configuration is deemed infeasible, the seed of the heuristic solver can be changed to find another configuration. The single configuration found as an example in this paper seems to be feasible. However, in practice, more experiments are needed to list layout alternatives and choose the one with the least transformation costs and/or the best suitability with respect to other architectural criteria.
- The proposed methodology bases the reconfiguration problems on a completely discretized and modularized design space, and the proposed algorithm is reasonably fast, it would be theoretically feasible to dissect the spatial units into smaller units and generalize the method to broaden the application areas and the versatility of the method for incorporating more diverse validity constraints.

Integrating QAP into computational design workflows can, to say the least, provides awareness of the logistics performance of the building in terms of the expected walking time for personnel, and in that sense, it can even be used as an informative tool for conceptual design of new buildings as well as re-designing existing buildings.

7. Limitations & future work

The method presented in this paper is only suitable for reconfiguration of existing buildings, especially because it requires computing the distance between available locations for computing the main objective function. Even though most hospital buildings have a regular structural grid, it must be noted that our proposed way of dissecting departments into modular areal units is only feasible on such highly modular and regularly structured buildings. It must be noted that in our problem formulation we consider all facilities to be accommodatable in all available locations, while in reality there might be facilities that can only be accommodated in certain locations due to particular technical requirements. While this constraint is handled by excluding a list of fixed facilities, we have disregarded the exchangeability of other departments. Our methodology does not take the contiguity constraints into account explicitly as hard constraints, e.g. in cases where we split a facility into 2 or 3 facilities to fit it into our modularized spaces. We cannot enforce the new units to stay contiguous/adjacent to each other during the optimization process; however, by adding extra closeness ratings in between the split parts, we relax such constraints and add them to the objective function effectively. Moreover, this limitation can also be considered in another way: that the obtained results, which may not strictly entail the initially conceived contiguity, can be used to reflect on the programme of requirements and consider revising it, e.g. considering two Cardiology departments if a significant expected travel-time saving can be made by splitting it into two departments. This can be observed from the objective function. As future work, limitations of the proposed workflow to the existing buildings can be addressed by modifying the problem-formulation, for instance by a Mixed-Integer Programming formulation of the hospital layout problem. More advanced optimization algorithms for solving the QAP can be proposed e.g. populated version of the proposed algorithm. The geodesic distances computed on the voxelated corridors are currently more accurate than Euclidean distance but still quite simplistic in that they do not consider the cost of waiting times for the elevators, nor do they differentiate between going downstairs and upstairs with distances on the same level. In general, instead of measuring distance in meters, it would be more general to measure distance in travel time/effort, to also account for path complexity for the visitors, for instance by using the Easiest Paths weighting [57] Corridor Allocation Problem (CAP) [58] can be added, which has the same fitness structure with the QAP but with an extra decision on locating the facilities on either sides of a corridor. Finally, the proposed layout optimization tool can be potentially extended to solve a Multi-Objective QAP [59,60] in further versions.

CRediT authorship contribution statement

Cemre Cubukcuoglu: Conceptualization, Methodology, Software, QAP solver, Writing – original draft. **Pirouz Nourian:** Conceptualization, Methodology, Software, Geodesic Distance, Slider Update in QAP solver, Writing – review & editing, Supervision. **M. Fatih Tasgetiren:** Conceptualization, Methodology, Software, QAP solver, Writing – review & editing, Supervision. **I. Sevil Sariyildiz:** Conceptualization, Methodology, Supervision. **Shervin Azadi:** Software, Geodesic Distance.

Declaration of competing interest

The authors declare that there is no conflict of interest in this paper.

Appendix

Appendix-A

Table 1
A List of Spaces with a needed number of boxes (modular units of roughly 100 m^2)

Facility Name	Number of Boxes Needed
NEUROLOGY	1
OBSTETRICS	1
INTERNAL MEDICINE	1
CARDIOLOGY	1
PEDIATRICS	1
GASTROENTEROLOGY	2
DIAGNOSTIC UNITS	2
LABORATORIES	2
ADMINISTRATION-1	1
ORTHOPEDY	1
PHYSIOTHERAPY	2
HEMODIALYSIS	2
INFECTION DISEASES	2
SURGERY ICU	2
GENERAL ICU	1
PEDIATRY ICU	1
ICU WAITING AREA	1
ENT	1
UROLOGY	1
EYE DISEASES	2
BRAIN&GENERAL SURGERY	1
DERMATOLOGY	1
INPATIENT-1	2
DELIVERY ROOMS	3
OPERATION THEATRE	5
ADMINISTRATION-2	4
CONFERENCE HALL	3
LIBRARY	1
INPATIENT-2	4
INPATIENT-3	4
INPATIENT-4	4
KITCHEN	2
DINING HALL	2
Total number of locations (modular units)	64

Appendix-B

The table below is a RELChart produced by collating expert interviews, site visits, design guidelines/standards, and recommendations from the scientific literature.

		NEUROLOGY	OBSTETRICS	INTERNAL MEDICINE	CARDIOLOGY	PEDIATRICS	GASTROENTOLOGY-1	GASTROENTOLOGY-2	DIAGNOSTIC UNITS-1	DIAGNOSTIC UNITS-2	LABORATORIES-1	LABORATORIES-2	ADMINISTRATION-1	ORTHOPEDY	PHYSIOTHERAPHY-1	PHYSIOTHERAPHY-2	HEMODIALYSIS-1	HEMODIALYSIS-2	INFECTION DISEASES-1	INFECTION DISEASES-2	SURGERY ICU-1	SURGERY ICU-2	GENERAL ICU	PEDIATRY ICU	ICU WAITING AREA	ENT	UROLOGY	EYE DISEASES-1	EYE DISEASES-2	BRAIN& GENERAL SURGERY	DERMATOLOGY	INPATIENT-1-1	INPATIENT-1-2	DELIVERY ROOMS-1	DELIVERY ROOMS-2	DELIVERY ROOMS-3	OPERATION THEATRE-1	OPERATION THEATRE-2	OPERATION THEATRE-3	OPERATION THEATRE-4	OPERATION THEATRE-5	ADMINISTRATION-2-1	ADMINISTRATION-2-2	ADMINISTRATION-2-3	ADMINISTRATION-2-4	CONFERENCE HALL-1	CONFERENCE HALL-2	CONFERENCE HALL-3	LIBRARY	INPATIENT-2-1	INPATIENT-2-2	INPATIENT-2-3	INPATIENT-2-4	INPATIENT-3-1	INPATIENT-3-2	INPATIENT-3-3	INPATIENT-3-4	INPATIENT-4-1	INPATIENT-4-2	INPATIENT-4-3	INPATIENT-4-4	KITCHEN-1	KITCHEN-2	DINING HALL-1	DINING HALL-2
0	NEUROLOGY	0	100	100	100	0	100	100	75	75	50	50	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	OBSTETRICS	100	0	100	100	0	100	100	75	75	50	50	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	INTERNAL MEDICINE	100	100	0	100	0	100	100	75	75	50	50	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	CARDIOLOGY	100	100	100	0	0	100	100	75	75	50	50	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	PEDIATRICS	0	0	0	0	0	100	100	75	75	50	50	50	0	100	100	0	0	100	100	0	0	0	100	100	0	0	0	0	0	100	0	0	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	GASTROENTOLOGY-1	100	100	100	100	100	0	100	37	37	37	25	25	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	GASTROENTOLOGY-2	100	100	100	100	100	100	0	37	37	37	25	25	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	DIAGNOSTIC UNITS-1	75	75	75	75	75	37	37	0	100	0	0	0	25	25	25	0	0	25	25	37	37	37	37	37	25	25	25	25	25	25	37	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	37	37	37	37	37	37	37	37	37	37	37	0	0	0	0
8	DIAGNOSTIC UNITS-2	75	75	75	75	75	37	37	100	0	0	0	0	25	25	25	0	0	25	25	37	37	37	37	37	25	25	25	25	25	25	37	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	37	37	37	37	37	37	37	37	37	37	37	0	0	0	0
9	LABORATORIES-1	50	50	50	50	50	37	37	0	0	0	100	0	25	25	25	0	0	25	25	37	37	37	37	37	25	25	25	25	25	25	37	37	0	0	0	25	25	25	25	25	0	0	0	0	0	0	0	0	25	25	25	25	25	25	25	25	25	25	25	25	0	0	0	0
10	LABORATORIES-2	50	50	50	50	50	25	25	0	0	100	0	0	25	25	25	0	0	25	25	37	37	37	37	37	25	25	25	25	25	25	37	37	0	0	0	25	25	25	25	25	0	0	0	0	0	0	0	0	25	25	25	25	25	25	25	25	25	25	25	25	0	0	0	0
11	ADMINISTRATION-1	0	0	0	0	50	25	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	25	25	25	25	25	25	25	25	25	25	25	0	0	0	0
12	ORTHOPEDY	100	100	100	100	0	0	0	25	25	25	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	PHYSIOTHERAPHY-1	0	0	0	0	100	50	50	25	25	25	25	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	PHYSIOTHERAPHY-2	0	0	0	0	100	0	0	25	25	25	25	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	HEMODIALYSIS-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	HEMODIALYSIS-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	INFECTION DISEASES-1	0	0	0	0	100	0	0	25	25	25	25	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	INFECTION DISEASES-2	0	0	0	0	100	0	0	25	25	25	25	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	SURGERY ICU-1	0	0	0	0	0	0	0	37	37	37	37	0	0	0	0	0	0	0	0	0	100	50	50	50	0	0	0	0	0	0	25	25	0	0	0	50	50	50	50	50	0	0	0	0	0	0	0	0	25	25	25	25	25	25	25	25	25	25	25	25	0	0	0	0
20	SURGERY ICU-2	0	0	0	0	0	0	0	37	37	37	37	0	0	0	0	0	0	0	0	100	0	50	50	50	0	0	0	0	0	0	25	25	0	0	0	50	50	50	50	50	0	0	0	0	0	0	0	0	25	25	25	25	25	25	25	25	25	25	25	25	0	0	0	0
21	GENERAL ICU	0	0	0	0	0	0	0	37	37	37	37	0	0	0	0	0	0	0	0	50	50	0	100	100	0	0	0	0	0	0	50	50	0	0	0	100	100	100	100	100	0	0	0	0	0	0	0	0	50	50	50	50	50	50	50	50	50	50	50	50	0	0	0	0
22	PEDIATRY ICU	0	0	0	0	100	0	0	37	37	37	37	0	0	0	0	0	0	0	0	50	50	100	0	100	0	0	0	0	0	0	50	50	100	100	100	100	100	100	100	100	0	0	0	0	0	0	0	0	50	50	50	50	50	50	50	50	50	50	50	50	0	0	0	0
23	ICU WAITING AREA	0	0	0	0	100	0	0	37	37	37	37	0	0	0	0	0	0	0	0	50	50	100	100	0	0	0	0	0	0	0	50	50	0	0	0	100	100	100	100	100	0	0	0	0	0	0	0	0	50	50	50	50	50	50	50	50	50	50	50	50	0	0	0	0
24	ENT	0	0	0	0	0	0	0	25	25	25	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	UROLOGY	0	0	0	0	0	0	0	25	25	25	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	EYE DISEASES-1	0	0	0	0	0	0	0	25	25	25	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	EYE DISEASES-2	0	0	0	0	0	0	0	25	25	25	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	BRAIN& GENERAL SURGERY	100	100	100	100	0	0	0	25	25	25	25	0	100	50	50	50	50	50	50	0	0	0	0	0	100	100	50	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	DERMATOLOGY	0	0	0	0	100	50	50	25	25	25	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	INPATIENT-1-1	0	0	0	0	0	50	50	37	37	37	37	0	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	0	100	0	0	0	50	50	50	50	50	12	12	12	12	0	0	0	0	12	12	12	12	12	12	12	12	12	12	12	12	37	37	0	0
31	INPATIENT-1-2	0	0	0	0	0	0	0	37	37	37	37	0	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	100	0	0	0	0	50	50	50	50	50	12	12	12	12	0	0	0	0	12	12	12	12	12	12	12	12	12	12	12	12	37	37	0	0
32	DELIVERY ROOMS-1	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	100	100	16	16	16	16	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	DELIVERY ROOMS-2	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	100	0	100	16	16	16	16	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	DELIVERY ROOMS-3	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	100	100	0	16	16	16	16	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	OPERATION THEATRE-1	0	0	0	0	0	0	0	0	0	25	25	0	0	0	0	0	0	0	0	50	50	100	100	100	0	0	0	0	0	0	50	50	16	16	16	0	100	100	100	100	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	0	0	0	0
36	OPERATION THEATRE-2	0	0	0	0	0	0	0	0	0	25	25	0	0	0	0	0	0	0	0	50	50	100	100	100	0	0	0	0	0	0	50	50	16	16	16	100	0	100	100	100	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	0	0	0	0
37	OPERATION THEATRE-3	0	0	0	0	0	0	0	0	0	25	25	0	0	0	0	0	0	0	0	50	50	100	100	100	0	0	0	0	0	0	50	50	16	16	16	100	100	0	100	100	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	0	0	0	0
38	OPERATION THEATRE-4	0	0	0	0	0	0	0	0	0	25	25	0	0	0	0	0	0	0	0	50	50	100	100	100	0	0	0	0	0	0	50	50	16	16	16	100	100	100	0	100	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	0	0	0	0
39	OPERATION THEATRE-5	0	0	0	0	0	0	0	0	0	25	25	0	0	0	0	0	0	0	0	50	50	100	100	100	0	0	0	0	0	0	50	50	16	16	16	100	100	100	100	0	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	0	0	0	0
40	ADMINISTRATION-2-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	12	0	0	0	0	0	0	0	0	0	100	100	100	20	20	20	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	ADMINISTRATION-2-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	12	0	0	0	0	0	0	0	0	100	0	100	100	20	20	20	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	ADMINISTRATION-2-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	12	0	0	0	0	0	0	0	0	100	100	0	100	20	20	20	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	ADMINISTRATION-2-4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	12	0	0	0	0	0	0	0	0	100	100	100	0	20	20	20	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	CONFERENCE HALL-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	20	0	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	CONFERENCE HALL-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	20	100	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	CONFERENCE HALL-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	20	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	LIBRARY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	INPATIENT-2-1	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	0	100	100	100	25	25	25	25	25	25	25	25	18	18	0	0
49	INPATIENT-2-2	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	100	0	100	100	25	25	25	25	25	25	25	25	18	18	0	0
50	INPATIENT-2-3	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	100	100	0	100	25	25	25	25	25	25	25	25	18	18	0	0
51	INPATIENT-2-4	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	100	100	100	0	100	100	100	100	25	25	25	25	18	18	0	0
52	INPATIENT-3-1	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	25	25	25	100	0	100	100	100	25	25	25	25	18	18	0	0
53	INPATIENT-3-2	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	25	25	25	100	100	0	100	100	25	25	25	25	18	18	0	0
54	INPATIENT-3-3	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	25	25	25	100	100	100	0	100	25	25	25	25	18	18	0	0
55	INPATIENT-3-4	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	25	25	25	100	100	100	100	0	100	100	100	100	18	18	0	0
56	INPATIENT-4-1	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	25	25	25	25	25	25	25	100	0	100	100	100	18	18	0	0
57	INPATIENT-4-2	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	25	25	25	25	25	25	25	100	100	0	100	100	18	18	0	0
58	INPATIENT-4-3	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	25	25	25	25	25	25	25	100	100	100	0	100	18	18	0	0
59	INPATIENT-4-4	0	0	0	0	0	0	0	37	37	25	25	25	0	0	0	0	0	0	0	25	25	50	50	50	0	0	0	0	0	0	12	12	0	0	0	20	20	20	20	20	0	0	0	0	0	0	0	0	25	25	25	25	25	25	25	100	100	100	100	0	18	18	0	0
60	KITCHEN-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	18	18	18	18	18	18	18	18	18	18	18	0	100	50	50
61	KITCHEN-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	18	18	18	18	18	18	18	18	18	18	18	100	0	100	50
62	DINING HALL-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	100	0	100
63	DINING HALL-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	50	100	0

Appendix-C

Table 3
The Existing Assignment of the Case Study Hospital

Location Index	The Existing Assignment		The Proposed Assignment	
	Facility Index	Facility Name	Facility Index	Facility Name
0	44	CONFERENCE HALL-1	56	INPATIENT-4-1
1	35	OPERATION THEATRE-1	19	SURGERY ICU-1
2	22	PEDIATRY ICU	29	DERMATOLOGY
3	7	DIAGNOSTIC UNITS-1	30	INPATIENT-1-1
4	8	DIAGNOSTIC UNITS-2	61	KITCHEN-2
5	3	CARDIOLOGY	15	HEMODIALYSIS-1
6	4	PEDIATRICS	25	UROLOGY
7	2	INTERNAL MEDICINE	50	INPATIENT-2-3
8	1	OBSTETRICS	48	INPATIENT-2-1
9	0	NEUROLOGY	55	INPATIENT-3-4
10	5	GASTROENTOLOGY-1	49	INPATIENT-2-2
11	9	LABORATORIES-1	32	DELIVERY ROOMS-1
12	6	GASTROENTEROLOGY-2	2	INTERNAL MEDICINE
13	10	LABORATORIES-2	54	INPATIENT-3-3
14	15	HEMODIALYSIS-1	60	KITCHEN-1
15	16	HEMODIALYSIS-2	47	LIBRARY
16	11	ADMINISTRATION-1	13	PHYSIOTHERAPHY-1
17	12	ORTHOPEDY	63	DINING HALL-2
18	13	PHYSIOTHERAPHY-1	57	INPATIENT-4-2
19	14	PHYSIOTHERAPY-2	6	GASTROENTEROLOGY-2
20	19	SURGERY ICU-1	4	PEDIATRICS
21	20	SURGERY ICU-2	17	INFECTION DISEASES-1
22	21	GENERAL ICU	3	CARDIOLOGY
23	17	INFECTION DISEASES-1	31	INPATIENT-1-2
24	18	INFECTION DISEASES-2	44	CONFERENCE HALL-1
25	23	ICU WAITING AREA	52	INPATIENT-3-1
26	30	INPATIENT-1-1	23	ICU WAITING AREA
27	31	INPATIENT-1-2	59	INPATIENT-4-4
28	26	EYE DISEASES-1	1	OBSTETRICS
29	28	BRAIN&GENERAL SURGERY	0	NEUROLOGY
30	27	EYE DISEASES-2	41	ADMINISTRATION-2-2
31	25	UROLOGY	9	LABORATORIES-1
32	24	ENT	26	EYE DISEASES-1
33	36	OPERATION THEATRE-2	45	CONFERENCE HALL-2
34	37	OPERATION THEATRE-3	7	DIAGNOSTIC UNITS-1
35	38	OPERATION THEATRE-4	12	ORTHOPEDY
36	39	OPERATION THEATRE-5	46	CONFERENCE HALL-3
37	32	DELIVERY ROOMS-1	11	ADMINISTRATION-1
38	29	DERMATOLOGY	40	ADMINISTRATION-2-1
39	33	DELIVERY ROOMS-2	24	ENT
40	34	DELIVERY ROOMS-3	62	DINING HALL-1
41	48	INPATIENT-2-1	28	BRAIN& GENERAL SURGERY
42	49	INPATIENT-2-2	51	INPATIENT-2-4
43	45	CONFERENCE HALL-2	35	OPERATION THEATRE-1
44	50	INPATIENT-2-3	10	LABORATORIES-2
45	47	LIBRARY	14	PHYSIOTHERAPY-2
46	51	INPATIENT-2-4	39	OPERATION THEATRE-5
47	46	CONFERENCE HALL-3	16	HEMODIALYSIS-2
48	52	INPATIENT-3-1	37	OPERATION THEATRE-3
49	53	INPATIENT-3-2	22	PEDIATRY ICU
50	54	INPATIENT-3-3	34	DELIVERY ROOMS-3
51	55	INPATIENT-3-4	36	OPERATION THEATRE-2
52	56	INPATIENT-4-1	38	OPERATION THEATRE-4
53	57	INPATIENT-4-2	20	SURGERY ICU-2
54	58	INPATIENT-4-3	43	ADMINISTRATION-2-4
55	59	INPATIENT-4-4	53	INPATIENT-3-2
56	60	KITCHEN-1	8	DIAGNOSTIC UNITS-2
57	61	KITCHEN-2	21	GENERAL ICU
58	62	DINING HALL-1	33	DELIVERY ROOMS-2
59	63	DINING HALL-2	5	GASTROENTEROLOGY-1
60	40	ADMINISTRATION-2-1	42	ADMINISTRATION-2-3
61	41	ADMINISTRATION-2-2	27	EYE DISEASES-2
62	42	ADMINISTRATION-2-3	58	INPATIENT-4-3
63	43	ADMINISTRATION-2-4	18	INFECTION DISEASES-2

Appendix-D

Table 4
Nomenclature

Notation	Explanation
QAP	Quadratic Assignment Problem
C#	C sharp programming language
GH	Grasshopper
CAD	Computer Aided Design
ILS	Iterated Local Search
FLP	Facility Layout Planning
STEP	A sample test-pairwise exchange heuristic procedure
GRASP	Greedy randomized adaptive search procedure
MSG	Modified sub-gradient
GA	Genetic algorithm
SA	Simulated annealing
TS	Tabu search
EGD	Extended great deluge
OT	Operating Theatre
2D	Two dimensional
3D	Three dimensional
OR	Operations Research
QAPLIB	A Quadratic Assignment Problem Library
CAP	Corridor Allocation Problem
A*	A-star
π	A vector of integers denoting a permutation of facilities at a moment in time
$T\pi i,\pi j$	The [transportation] flow between a permuted facility $\pi i = k$, and another permuted facility $\pi j = l$
Pn	The set of all permutations
Di,j	The distance between location i and j
t_max	Maximum number of trial times
n	Number of facilities & locations
CRAFT	Computerized Relative Allocation of Facilities Technique
COFAD	COmputerized FAcilities Design
CORELAP	Computerized Relationship Layout Planning

References

[1] Space Planning Basics, fourth ed., Wiley, (n.d.), https://www.wiley.com/en-gb/Space+Planning+Basics%2C+4th+Edition-p-9781118882009. (Accessed 13 December 2019). accessed.

[2] A. Chraibi, S. Kharraja, I.H. Osman, O. Elbeqqali, A mixed integer programming formulation for solving operating theatre layout problem: a multi-goal approach, in: Proc. - IESM Conf. Oct. 2013, Rabat, Morocco, 2013.

[3] R. Ulrich, X. Quan, H. Systems, C. Architecture, A. Texas, The role of the physical environment in the hospital of the 21 st Century : a once-in-a-lifetime opportunity, Environment 439 (2004) 69.

[4] L.D. Burgio, B.T. Engel, A. Hawkins, K. McCormick, A. Scheve, A descriptive analysis of nursing staff behaviors in a teaching nursing home: differences among NAs, LPNs, and RNs, Gerontol. 30 (1990) 107–112.

[5] M.M. Shepley, K. Davies, Nursing unit configuration and its relationship to noise and nurse walking behavior: an AIDS/HIV unit case study, AIA Acad. J. 6 (2003) 12–14.

[6] H. Pourvaziri, H. Pierreval, Dynamic facility layout problem based on open queuing network theory, Eur. J. Oper. Res. 259 (2017) 538–553, https://doi.org/10.1016/j.ejor.2016.11.011.

[7] T.C. Koopmans, M. Beckmann, Assignment problems and the location of economic activities, Econom. J. Econom. Soc. (1957) 53–76.

[8] B.K. Kaku, G.L. Thompson, I. Baybars, A heuristic method for the multi-story layout problem, Eur. J. Oper. Res. 37 (1988) 384–397, https://doi.org/10.1016/0377-2217(88)90202-0.

[9] B.K. Kaku, R. Rachamadugu, Layout design for flexible manufacturing systems, Eur. J. Oper. Res. 57 (1992) 224–230, https://doi.org/10.1016/0377-2217(92)90044-A.

[10] M.J. Rosenblatt, B. Golany, A distance assignment approach to the facility layout problem, Eur. J. Oper. Res. 57 (1992) 253–270, https://doi.org/10.1016/0377-2217(92)90047-D.

[11] W.J. Li, J.M. Smith, An algorithm for quadratic assignment problems, Eur. J. Oper. Res. 81 (1995) 205–216, https://doi.org/10.1016/0377-2217(93)E0162-Q.

[12] T.L. Urban, Solution procedures for the dynamic facility layout problem, Ann. Oper. Res. 76 (1998) 323–342.

[13] A.S. Ramkumar, S.G. Ponnambalam, N. Jawahar, A new iterated fast local search heuristic for solving QAP formulation in facility layout design, Robot. Comput. Integrated Manuf. 25 (2009) 620–629, https://doi.org/10.1016/j.rcim.2008.03.022.

[14] C.L. Huntley, D.E. Brown, A parallel heuristic for quadratic assignment problems, Comput. Oper. Res. 18 (1991) 275–289, https://doi.org/10.1016/0305-0548(91)90029-Q.

[15] P.P.C. Yip, Y.-H. Pao, A guided evolutionary simulated annealing approach to the quadratic assignment problem, IEEE Trans. Syst. Man. Cybern. 24 (1994) 1383–1387, https://doi.org/10.1109/21.310514.

[16] J.A. Bland, G.P. Dawson, Large-scale layout of facilities using a heuristic hybrid algorithm, Appl. Math. Model. 18 (1994) 500–503, https://doi.org/10.1016/0307-904X(94)90328-X.

[17] C. Wen-Chyuan, C. Chi, Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation, Eur. J. Oper. Res. 2217 (1998) 457–488, https://doi.org/10.1016/S0377-2217(97)00285-3.

[18] J.S. Kochhar, B.T. Foster, S.S. Heragu, Hope, A genetic algorithm for the unequal area facility layout problem, Comput. Oper. Res. 25 (1998) 583–594, https://doi.org/10.1016/S0305-0548(97)00100-7.

[19] W.-C. Chiang, Visual facility layout design system, Int. J. Prod. Res. 39 (2001) 1811–1836, https://doi.org/10.1080/00207540110035192.

[20] M. Solimanpur, P. Vrat, R. Shankar, Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing, Eur. J. Oper. (2004).

[21] M. Nourelfath, N. Nahas, B. Montreuil, Coupling ant colony optimization and the extended great deluge algorithm for the discrete facility layout problem, Eng. Optim. 39 (2007) 953–968, https://doi.org/10.1080/03052150701551461.

[22] J.R. Jaramillo, A.R. McKendall Jr., The generalised machine layout problem, Int. J. Prod. Res. 48 (2010) 4845–4859, https://doi.org/10.1080/00207540903117840.

[23] G. Moslemipour, T.S. Lee, Intelligent design of a dynamic machine layout in uncertain environment of flexible manufacturing systems, J. Intell. Manuf. 23 (2012) 1849–1860, https://doi.org/10.1007/s10845-010-0499-8.

[24] S.P. Singh, R.R.K. Sharma, A review of different approaches to the facility layout problems, Int. J. Adv. Manuf. Technol. 30 (2006) 425–433, https://doi.org/10.1007/s00170-005-0087-9.

[25] A.N. Elshafei, Hospital layout as a quadratic assignment problem, Oper. Res. Q. 28 (1977) 167–179.

[26] B.A. Murtagh, T.R. Jefferson, V. Sornprasit, A heuristic procedure for solving the quadratic assignment problem, Eur. J. Oper. Res. 9 (1982) 71–76, https://doi.org/10.1016/0377-2217(82)90013-3.

[27] T.W. Butler, K.R. Karwan, J.R. Sweigart, G.R. Reeves, An integrative model-based approach to hospital layout, IIE Trans. 24 (1992) 144–152, https://doi.org/10.1080/07408179208964211.

[28] P.M. Hahn, J. Krarup, A hospital facility layout problem finally solved, J. Intell. Manuf. 12 (2001) 487–496, https://doi.org/10.1023/A:1012252420779.

[29] I.C. Yeh, Architectural layout optimization using annealed neural network, Autom. ConStruct. 15 (2006) 531–539, https://doi.org/10.1016/j.autcon.2005.07.002.

[30] A. Chraibi, S. Kharraja, I.H. Osman, O. Elbeqqali, Multi-agent system for solving Dynamic Operating Theater facility layout problem, IFAC-PapersOnLine. 28 (2015) 1146–1151, https://doi.org/10.1016/j.ifacol.2015.06.238.

[31] S. Helber, D. Böhme, F. Oucherif, S. Lagershausen, S. Kasper, A hierarchical facility layout planning approach for large and complex hospitals, Flex. Serv. Manuf. J. 2 (2014) 5–29, https://doi.org/10.1007/s10696-015-9214-6.
[32] X. Zuo, B. Li, X. Huang, M. Zhou, C. Cheng, X. Zhao, Z. Liu, Optimizing hospital emergency department layout via multiobjective tabu search, IEEE Trans. Autom. Sci. Eng. 16 (2019) 1137–1147.
[33] G.C. Armour, E.S. Buffa, A heuristic algorithm and simulation approach to relative location of facilities, Manag. Sci. 9 (1963) 294–309.
[34] J.A. Tompkins, R. Reed Jr., An applied model for the facilities design problem, Int. J. Prod. Res. 14 (1976) 583–595.
[35] R.C. Lee, CORELAP-computerized relationship layout planning, Jour. Ind. Engg. 8 (1967) 195–200.
[36] C.E. Donaghey, V.F. Pire, Facilities layout with BLOCPLAN, Univ. Houst. TX Work. Pap., 1991.
[37] A. Abdulmawla, M. Bielik, P. Buš, M.-C. Chang, M. Dennemark, E. Fuchkina, Y. Miao, K. Knecht, R. Koenig, S. Schneider, DeCodingSpaces toolbox for grasshopper. Computational analysis and generation of street network, Plots and Buildings (2017).
[38] R. Schaffranek, SpiderWeb. https://www.food4rhino.com/app/spiderweb, 2012.
[39] P. Nourian, S. Rezvani, S. Sariyildiz, A syntactic architectural design methodology, in: Meet. Ninth Int. Sp. Syntax Symp, 2013. Seoul, Korea.
[40] Introduction to the C# Language and the, NET Framework | Microsoft Docs, (n.d.), https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework. (Accessed 22 January 2020). accessed.
[41] D. Rutten, Grasshopper3D, (2007).
[42] VEX language reference (n.d.), https://www.sidefx.com/docs/houdini/vex/lang.html. (Accessed 22 January 2020). accessed.
[43] Houdini - 3D modeling, animation, VFX, look development, lighting and rendering | SideFX (n.d.), https://www.sidefx.com/. (Accessed 22 January 2020). accessed.
[44] P. Nourian, R. Gonçalves, S. Zlatanova, K.A. Ohori, A. Vu Vo, Voxelization algorithms for geospatial applications: computational methods for voxelating spatial datasets of 3D city models containing 3D surface, curve and point data models, Methods (Duluth) 3 (2016) 69–86, https://doi.org/10.1016/j.mex.2016.01.001.
[45] S. Azadi, P. Nourian, topoGenesis: an open-source python library providing topological structures and functions for Generative Systems and Sciences. https://doi.org/10.5281/ZENODO.4006514, 2020.
[46] M.M. Gunal, A guide for building hospital simulation models, Heal. Syst. 1 (2012) 17–25.
[47] C. Cubukcuoglu, P. Nourian, I.S. Sariyildiz, M.F. Tasgetiren, A discrete event simulation procedure for validating programs of requirements: the case of hospital space planning, Software 12 (2020) 100539, https://doi.org/10.1016/j.softx.2020.100539.
[48] S. Laine, A topological approach to voxelization, Comput. Graph. Forum 32 (2013) 77–86, https://doi.org/10.1111/cgf.12153.
[49] A.S. Ramkumar, S.G. Ponnambalam, N. Jawahar, A new iterated fast local search heuristic for solving QAP formulation in facility layout design, Robot. Comput. Integrated Manuf. 25 (2009) 620–629.
[50] M.S. Hussin, T. Stützle, Hierarchical iterated local search for the quadratic assignment problem, in: Int. Work. Hybrid Metaheuristics, 2009, pp. 115–129.
[51] H.R. Lourenço, O.C. Martin, T. Stützle, Iterated local search: framework and applications, in: Handb. Metaheuristics, Springer, 2019, pp. 129–168.
[52] T. Stützle, Iterated local search for the quadratic assignment problem, Eur. J. Oper. Res. 174 (2006) 1519–1539.
[53] M.F. Tasgetiren, Q.-K. Pan, Y. Ozturkoglu, O.K. Cotur, Variable block insertion heuristic for the quadratic assignment problem, 2017 IEEE Congr. Evol. Comput. (2017) 1765–1770.
[54] R.E. Burkard, S.E. Karisch, F. Rendl, QAPLIB–a quadratic assignment problem library, J. Global Optim. 10 (1997) 391–403.
[55] M.F. Tasgetiren, Q.-K. Pan, P.N. Suganthan, I.E. Dizbay, Metaheuristic algorithms for the quadratic assignment problem, 2013 IEEE Symp. Comput. Intell. Prod. Logist. Syst. (2013) 131–137.
[56] OpenVDB (n.d.), https://www.openvdb.org/. (Accessed 22 January 2020). accessed.
[57] P. Nourian, F. van der Hoeven, S. Rezvani, S. Sariylidiz, Easiest paths for walking and cycling : combining syntactic and geographic analyses in studying walking and cycling mobility, Proc. 10th Int. Sp. Syntax Symp. (2015) 1–15.
[58] A.R.S. Amaral, The corridor allocation problem, Comput. Oper. Res. 39 (2012) 3325–3330, https://doi.org/10.1016/j.cor.2012.04.016.
[59] C. Sanhueza, F. Jiménez, R. Berretta, P. Moscato, PasMoQAP: a parallel asynchronous memetic algorithm for solving the Multi-Objective Quadratic Assignment Problem, in: Evol. Comput. (CEC), 2017 IEEE Congr, ., 2017, pp. 1103–1110.
[60] C. Cubukcuoglu, M.F. Tasgetiren, I.S. Sariyildiz, L. Gao, M. Kucukvar, A memetic algorithm for the bi-objective quadratic assignment problem, Procedia Manuf 39 (2019) 1215–1222.