publish-Ing.

Journal of Production Systems and Logistics
Volume 5 | Article 7
Published: November 2025

Using of a Metaheuristic Algorithm as an Approach to the Low-threshold Creation of a Functional Scheme for Hospitals – A Case Study

Yuhao Gao[1], Lena Wecken[2], Kunibert Lennerts[1], Peter Nyhuis[2]

[1] *Institute of Technology and Management in Construction, Karlsruhe Institute of Technology, Karlsruhe, Germany*
[2] *Institute of Production Systems and Logistics, Garbsen, Germany*

Abstract

This study proposes a function scheme optimization method based on genetic algorithm, which aims to optimize the spatial layout within the hospital, reduce logistics and labour costs, and improve resource utilisation efficiency. Hospital layout planning is modelled as a quadratic assignment problem (QAP). In this framework, the object flow intensity of each functional area and between them is identified by constructing functional patterns and transformed into a flow-oriented ideal layout diagram. Elite selection, roulette selection, uniform crossover and mutation operations are used in the genetic algorithm, while parameters including elite ratio, crossover probability and mutation probability are set to achieve layout optimisation. The algorithm ensures the generation of feasible layouts through constraints such as room overlap and building boundary constraints, which also provides compactness ratio calculations to measure spatial dispersion and support layout optimisation. The case study results show that the proposed algorithm can efficiently generate layout solutions that conform to the functional pattern, which helps to reduce transport distances, reduce cross-contamination risks, and improve spatial utilisation and operational response speed of hospital resources.

Keywords

hospital planning; function scheme; transport effort; quadratic assignment problem; genetic algorithm

1. Introduction

There are many reasons for planning a layout in a hospital: A fundamental layout concept must be created for a new hospital planning, an existing layout is not transformable and must be adapted to external drivers of change, or an existing layout causes long transport distances as well as times and must be adapted according to the process efficiency, to name just a few. In particular, a lack of process efficiency in hospital operations causes dissatisfaction among the hospital staff involved and patients, while also causing additional costs [1]. Hospitals everywhere are under great economic pressure, but particularly in Germany due to the dual financing system [2]. Hospital planners are therefore under great pressure to design layout concepts that meet operational requirements and are thus able to increase process efficiency. There are already approaches in the literature to minimizing travel distances, logistical processes, and resulting costs in order to optimize hospital layouts. There is a vast amount of literature on using quadratic assignment problems (QAP) to optimize hospital layouts considering various real-world constraints. Because of the complexity of hospital layout planning and the enormous solution space of structure elements locations, QAP is highly desirable to solve layout planning problems in hospitals. For example, CUBUKCUOGLU ET AL.

This Paper has been reviewed by the Certified Reviewer Community of publish-Ing. – 1 review – single blind

DOI: https://doi.org/10.15488/19023
ISSN: 2702-2587

developed a QAP based heuristic solver using the spatial network of a hospital building to compute geodesic distances instead of rectilinear distances [1]. ZUO ET AL. used QAP formulation to solve the layout problem in an emergency department using a multi objective tabu search [3]. HELBER ET AL. proposed a hierarchical layout planning for large and complex hospitals, in which they assign organisational units to locations using QAP, minimising transport efforts in the first stage. The second stage is about detailed positioning within individual locations [4]. CHRAIBI ET AL. proposed a multi-agent decision-making system based on QAP and mixed integer programming for large operating room departments [5]. These approaches are complex, so the hurdle for hospital planners to actually apply the approaches in practice is high. There is a lack of a low-threshold approach that is applicable for hospital planners. This problem has already been formulated by WECKEN ET AL., who propose an approach based on layout planning for factories as a solution [6]. In addition to the layout planning itself, this takes into account five further steps that must be taken first: Target definition, establishing the project basis, structural planning, creation of the function scheme and dimensioning. This approach allows to address all the target requirements, such as transformability, transparency or transport efficiency, that were set for the hospital or the layout concept during target definition. In order to explicitly meet the requirement of a layout concept that increases transport efficiency, particular attention must be paid to reducing transport distances when creating the function scheme. The planning of the function scheme is therefore a critical planning phase during layout planning, for which practical support for hospital planners is needed. This article presents an algorithm that can be used to create the function scheme in a practice-oriented way. It formulates the task as a QAP and sets the structural elements defined by Wecken et al. in relation to their flow intensities in an initial ideal spatial arrangement to one another. By taking up the structural elements, the generic applicability of the algorithm to a wide range of planning cases is ensured.

The paper first outlines the necessary fundamentals regarding structural elements in the hospital, the functional scheme and the QAP. Subsequently, the algorithm is presented and an application guide is described, which is intended to serve as a practical orientation for hospital planners. The application of the algorithm and its benefits are tested using an abstract example of a real hospital planning case. Therefore, the procedure for creating a functional diagram by a hospital expert is compared with that created by the algorithm.

2. Fundamentals

This section outlines the basics of structural elements in a hospital, the functional diagram and how it is created, as well as the GA and QAP, in order to provide an understanding of the following algorithm and its application.

2.1 Function Scheme

Figure 1 on the right shows an example of an ideal function scheme of a hospital, which illustrates the area units and their material flow intensity. It combines the area units into an ideal, flow-oriented image based on their relationships with each other. The function scheme is based on the preceding structural planning, as also visualized in Figure 1 on the left [7]. This includes all units of area for which an organizationally independent spatial area must be provided in the final layout. Particularly in the context of a hospital re-planning, it can be challenging to identify all the area units at this early planning stage. The structural elements of a hospital defined by WECKEN ET AL. [6] provide a generic description of the area units of a hospital system. They can thus be used as a suitable basis for planning the functional layout.

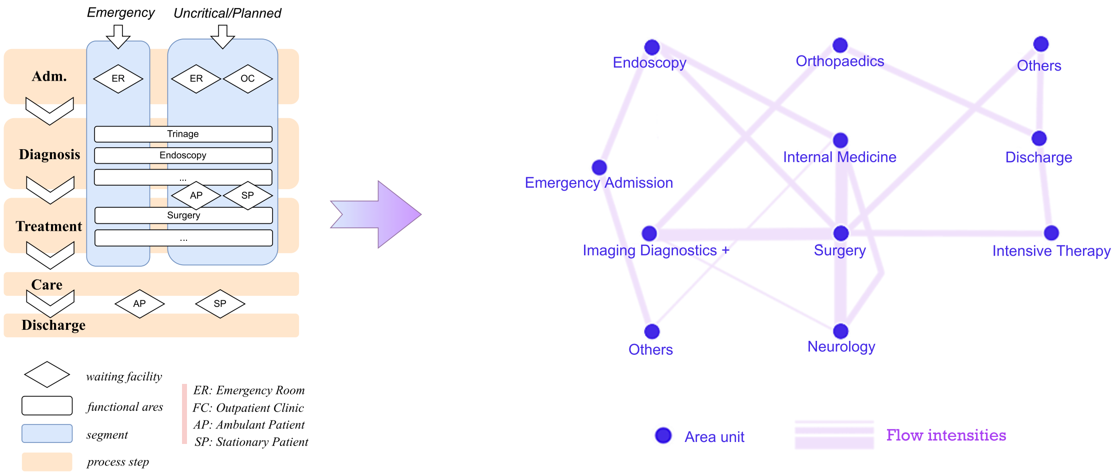

Figure 1: Generation of a Function Scheme from a Structure. Left: structural planning; Right: ideal function scheme

The process sequences can thus be derived from the function scheme and also initial indications for the spatial arrangement can result, but not yet the final spatial arrangement of the structural elements in the form of a realizable layout [8]. This is only possible after subsequent dimensioning as part of the layout planning [6].

2.2 Genetic Algorithm

Genetic algorithm (GA) is a robust population-based metaheuristic optimization algorithm inspired by the biological process of evolution, specifically the Darwinian theory of survival of the fittest in nature [9]. First proposed by J.H. HOLLAND in 1992, GAs are search-based algorithms grounded in the principles of natural selection and heredity [10]. The workflow of a GA begins with the initialization of a random population, often generated from a Gaussian distribution to enhance diversity. Each solution in this population represents a chromosome, composed of variables simulating genes, with the goal of distributing these solutions uniformly across the search space to improve the chances of finding global optimal regions [11]. GA employs four key operators: selection, crossover, mutation and elitism:

Selection: this operator draws inspiration from natural selection, where the fittest individuals are more likely to obtain resources and reproduce, thereby having higher probability passing on their genes to the next generation. Mirroring this concept, GA uses a roulette wheel mechanism to assign probabilities to individuals based on their fitness (objective) values, selecting them for creating next generation proportionally. Although this stochastic process gives less fit individuals a small chance of being chosen, it still allows their genes to contribute to future generations. This approach helps maintain population diversity, which is crucial to avoid prematurely narrowing the search space [11].

Crossover (Recombination): Once individuals are selected through a selection operator, they are used to generate offspring. In nature, this process involves combining the chromosomes from the genes of a male

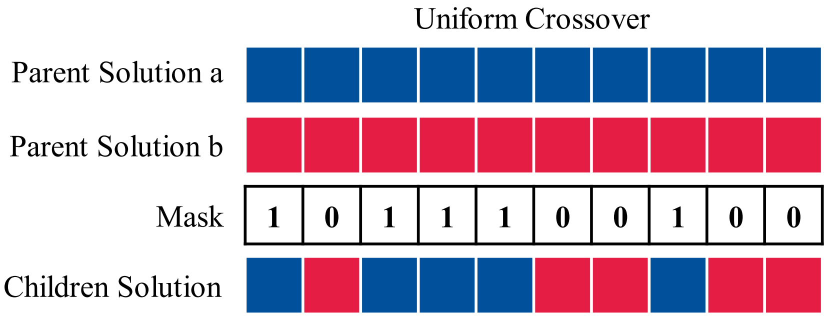

Figure 2: Uniform Crossover in Genetic Algorithm – Offspring solution is generated by selecting genes from two parent solutions based on a binary mask

and a female to create a new chromosome. The GA algorithm replicates this by merging two parent solutions, chosen by the roulette wheel, to produce two new offspring solutions. Various crossover techniques are documented in the literature, one of the commonly used crossover operators is uniform crossover shown in Figure 2, where each locus of two paired individuals is selected from either parent with equal probability. In uniform crossover, each position is treated individually, rather than in segments [12].

Mutation: The mutation operator involves altering one or more genes in the offspring solutions. In GA, the mutation rate is kept low to prevent the algorithm from devolving into a simple random search. This operator introduces additional randomness, helping to maintain population diversity. By preventing solutions from becoming too similar, it reduces the risk of the algorithm getting trapped in local optima. As shown in Figure 3, slight modifications occur in some randomly selected genes after the crossover (recombination) phase [11].

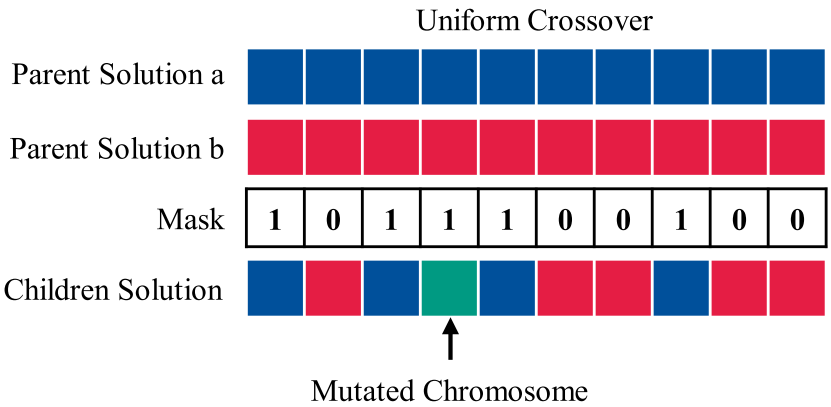

Figure 3: Mutation in Genetic Algorithm introduces random variation to enhance diversity

Elitism: Elitism is another widely used evolutionary operator [13] that involves preserving and directly transferring one or more of the best solutions to the next generation without alteration. The primary goal is to ensure these top solutions (elites) are not degraded during the application of crossover or mutation operators [11].

2.3 Quadratic Assignment Problem

The Quadratic Assignment Problem (QAP) is one of the most challenging problems in the NP-hard class. It has applications in a number of real-world domains, including facility siting, parallel and distributed computing, and combinatorial data analysis. Combinatorial optimisation problems, including the travelling salesman problem, maximum clique, and graph partitioning, can be formulated as QAPs [14].

The distances between locations, the demand flows between facilities and, in the general case, the costs of assigning facilities to locations are known. The international literature identifies the Quadratic Assignment Problem (QAP) as the problem of finding a minimum-cost assignment of facilities to locations, where costs are the sum of all possible distance-flow products [14].

3. Method

One of the key objectives in optimizing hospital layouts is to improve the efficiency of logistics and personnel movement, making flow optimization a crucial aspect of the planning process. By reducing the average distances between functional areas, the internal traffic flow within the hospital can be significantly enhanced. Flow optimization not only shortens transportation routes and reduces time and labor costs but also helps minimize the risk of cross-contamination. Additionally, optimizing the layout can decrease unnecessary congestion and increase the utilization of hospital resources. This paper formulates hospital layout planning as a Quadratic Assignment Problem (QAP) and proposes an elitism-based genetic algorithm approach to optimize internal flows. By adopting this method, hospitals can ensure effective connectivity between core areas while maximizing logistical efficiency, enabling quick responses and low-cost operations in a dynamic healthcare environment.

3.1 Function Scheme

This research investigates the optimization of spatial layouts within a specified environment, exemplified here by a hospital setting. To address the layout arrangement, the study applies the Quadratic Assignment Problem (QAP) framework. In this problem, a set of n rooms [1], R, and a set of n locations, L are defined to represent the hospital layout in a modularized concept. The variable f_{ij} represents the flow volume between rooms i and j, while d_{kl} is the city block distance between location k and location l, which is calculated by

$$d_{kl} = |x_k - x_l| + |y_k - y_l| \quad (1)$$

The layout area is modelled as a rectangular grid, with its boundaries specified by the coordinates $C_g = [x_{g,min}, x_{g,max}, y_{g,min}\ y_{g,max}]$. The goal is to allocate each facility to a position within this grid in a way that minimizes the total cost, which depends on both the distance and flow between facilities:

$$Minimize \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} f_{ij} \cdot d_{kl} \quad (2)$$

In order to find realizable solution in the real world, constraints should be applied to the problem. To prevent room overlap, the following condition must be met:

$$Overlap(C_i, C_j) = 0 \quad \forall i \neq j \quad (3)$$

where C_i and C_j are the coordinates of rooms i and j, respectively.

Additionally, each room must stay within the building boundaries, ensuring that its occupied area is contained within the layout limits. For room i, this is expressed as:

$$x_{g,min} \leq x_{i,min} \leq x_{i,max} \leq x_{g,max} \quad (4)$$

$$y_{g,min} \leq y_{i,min} \leq y_{i,max} \leq y_{g,max} \quad (5)$$

[1] Rooms are defined as any area or section of a hospital, not necessarily an enclosed space.

Compactness in layout design refers to maximizing the utilization of available space by clustering rooms to minimize unused areas and reduce unnecessary transport routes. To evaluate this aspect, a compactness ratio cr is defined and calculated as follows:

$$cr = \frac{\sum A(i)}{max(y_{i,max} - y_{i,min}) \times min(x_{i,max} - x_{i,min})} \tag{6}$$

where $\sum A(i)$ represents the total area covered by the rooms.

This metric indicates the degree of spatial dispersion among the rooms. A lower compactness ratio suggests that the rooms are closely packed, leaving minimal unused space in the layout. Consequently, the objective function is adjusted as follows:

$$Minimize \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} f_{ij} \cdot d_{kl} + w \times cr \tag{7}$$

where w is the weight of cr.

3.2 Model Development

Since QAP is NP-hard, the genetic algorithm (GA) is used as the solution approach. Genetic algorithm is a robust optimization algorithm and operates as a population-based metaheuristic algorithm.

The genetic algorithm in this paper is implemented in Python programming language on a machine with an NVIDIA RTX 3060 GPU, 16GB RAM, and an AMD Ryzen 7 6800H with Radeon Graphics CPU. All experiments were conducted on this setup to ensure consistency in performance measurement. The integrated development environment (IDE) is Jupyter Notebook in Anaconda. The version of Python is 3.11. The packages used in the code are genetic algorithm in PyPI, Matplotlib, SciPy, NumPy and Pandas.

The movement frequency f_{ij} is obtained through flow matrix, and the weight w of cr is 5000. For each population, check whether the constraints are violated, that is, whether the rooms overlap or exceed the building range. The population size is set to 500, the maximum number of iterations is 3000, and the selection operator consists of two parts: elite selection and roulette selection. The best individual will be directly selected as the parent individual. This method retains the best genes. The remaining parents are selected from the population through the roulette selection mechanism. The individuals with higher fitness are more likely to be selected. The type of crossover operator is uniform crossover, with a crossover probability of 0.5, a mutation probability of 0.1, and an elite ratio of 0.09. In addition, early stopping criteria are set. If the fitness score does not improve within 1000 iterations, the algorithm iteration is terminated.

Genetic algorithms are characterized by many parameters. The selection and adjustment of each parameter may affect the convergence speed of the algorithm and the quality of the solution. The following tuning suggestions are proposed for the primary parameters:

- Maximum number of loops: the genetic algorithm's termination criterion. Users can set the appropriate maximum number of loops according to the dimension, complexity and population size of the problem.
- Population size: determines the number of solutions generated at each iteration. A larger population can increase the exploration space but may also increase the computational cost. A smaller population may cause the algorithm to converge to a local optimal solution early.
- Mutation probability: determines the probability that each gene in each individual solution will be replaced by a random value. The smaller the mutation probability, the more likely it is that the solution will fall within the local optimum. If the mutation probability is too high, the algorithm becomes a random search.

- Elite Ratio: The number of elites in the population. The few most adaptable individuals are identified as elites and are directly retained in the iteration. Genetic algorithms with elitism are also called elite GAs.
- Crossover probability: Determines the probability that an existing solution will pass its genome to its offspring. The crossover rate is generally set to a higher value, such as 70% or 80%, to enhance the exploration of the search space. If the crossover rate is too high, it may lead to a decrease in the diversity of solutions.

The algorithm in this paper has the following steps:

- Randomly initialize a population P_0 consisting of with size N.
- Evaluate the fitness of each chromosome in the population P_0.
- Calculate the fitness using Eq. 4.
- Check for the constraints and in case of violation penalize the fitness value.
- Perform tournament selection.
- Perform crossover and mutation operations on the new generated parent population to create a new child population.
- Combine the original population with the newly generated offspring to form a combined population.
- Repeat steps until the maximum number of generations is reached.

4. Application guide

In order to apply the above algorithm for creating a function scheme, it was functionalized using Python. This section presents the procedure by which the algorithm can be applied to a specific planning example by hospital planners in practice. This ensures that the solution approach is easily accessible. The Python code is available in GitHub (https://github.com/Yuhao-27/DFG_MedFAP).

4.1 Data preparation

In order for the algorithm to produce results, it must be fed with flow matrices. Flow matrices contain data on the area units to be planned and the flow intensities between them. In most cases, the flow matrices themselves will not be available at the beginning of the hospital planning project and must be generated using flow data, see Figure 4. Flow data is recorded, if at all, in different forms and levels of detail depending on the hospital. This means that a large number of different flow objects are recorded in a system and thus in a data set. In order to create manageable and meaningful flow matrices from this data, the flow objects must first be classified (see Figure 4) using the flow object classes included in Appendix. The classes were created based on DIN 13080 [15] and finalized in discussions with hospital planners. In addition to the flow objects, the pick-up and arrival locations in the original data set are often not available in the form necessary for further use. They are often at a very detailed level of observation, which is not useful for function scheme planning. The level of detail must therefore be reduced in a second step (see Figure 4) by assigning the individual pick-up and arrival locations to the area units or the structural elements defined by WECKEN ET AL. [6] and summarizing them accordingly. If no technically recorded flow data exists, it can be qualitatively determined with the help of interviews by specialized personnel. However, it is important to ensure that the group of interviewees is set up in such a way that all specialist areas are represented and that no flow relationships are later omitted from the function scheme. In addition, comparability of the flow intensities should be ensured in order to be able to subsequently place them in a realistic relation.

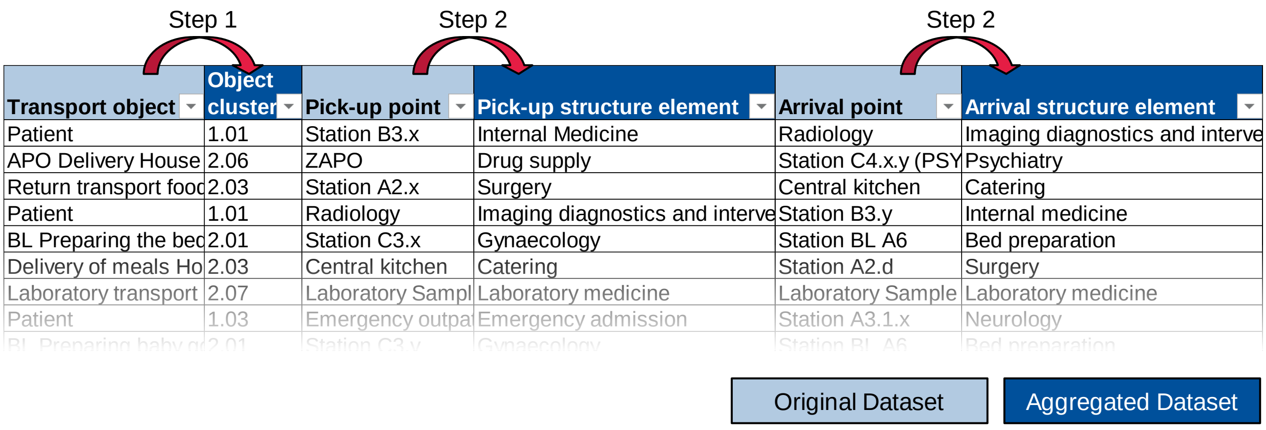

Figure 4: Preparation of Flow Data

In a third step, this flow information can then be cumulated for each transport object and transferred to flow matrices, see Figure 5.

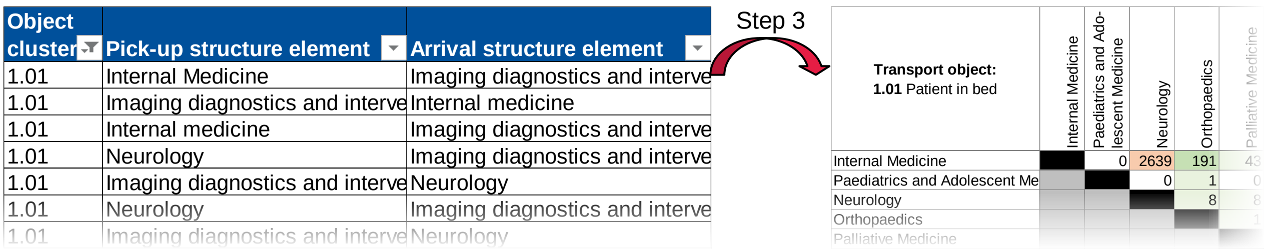

Figure 5: Generating the Flow Matrices

The final layout is evaluated based on the requirements defined as part of the target definition [6]. When creating the function scheme, it may therefore be sufficient to only consider the transport objects that influence the achievement of objectives. For example, the objective of layout planning may be to improve routes for patients. In this case, only transport object classes 1.01 to 1.04 are relevant for function scheme planning. If the objective is to improve the routes for patients in beds and wheelchairs and for transport staff at the same time, the flow matrices for transport object classes 1.01, 1.02 and 2.01 should be selected, for example. In a fourth step of data preparation, as exemplified in Figure 6, a selection of the flow matrices can thus be made.

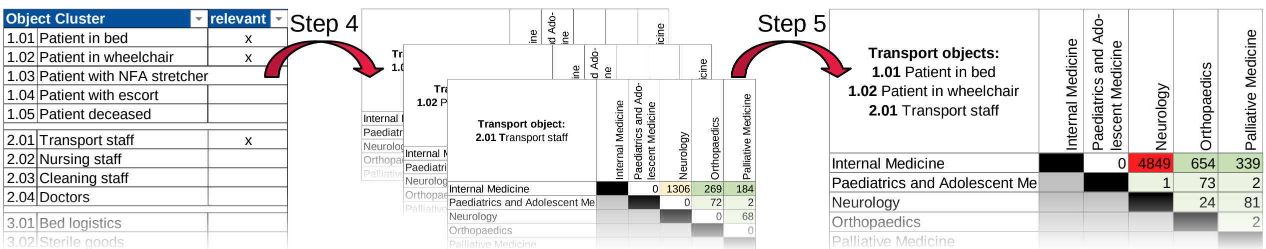

Figure 6: Aggregation of the Input-flow Matrix

Finally, in the fifth step of the data preparation, the flow matrices of the relevant transport object classes are to be aggregated into a flow matrix that can then be used to feed the algorithm. It is possible to further weight or manipulate the input data. Weighting can be useful, for example, if the target requirements have different priorities and this is to be taken into account. Manipulation, on the other hand, can be useful if it is known that only a proportion of the transports are included in the data due to problems in data collection. It is also possible that the actual transport costs are not reflected in the data because, for example, the personnel costs for patient transport in bed are often twice as high as for patient transport in a wheelchair, but both are included in the data.

4.2 Model Application

Firstly, the programming environment must be set up. This entails ensuring the installation of the requisite packages, including pandas, NumPy, and any other libraries essential for data processing and algorithmic execution. It is of the utmost importance to ensure that the requisite environment is configured correctly, as this is a prerequisite for the subsequent steps to be carried out in an optimal manner.

In the second step, relevant data should be imported. This involves loading the dataset from the Excel file into the Python environment. This may be accomplished through the utilization of the pandas library, particularly the “read_excel()” function. It is essential to provide the correct file path in order to ensure the accurate import of the data. Please refer to the screenshot, which illustrates the function run “read_excel()”. The user could then enter the path of the filename in the interface shown in Figure 7.

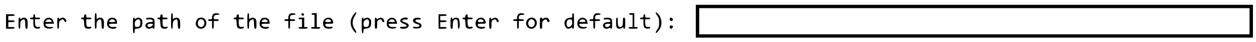

Figure 7: Interface for data importation

Once the requisite data has been prepared, the selected algorithm should be executed with the predefined parameters. It is essential to ensure that the algorithm is correctly configured and initialized in accordance with the requirements of the analysis. This stage represents the core computational process through which initial results are generated.

Following the execution of the algorithm, the results should be evaluated through the medium of visualization which is shown in Figure 8. The utilization of data visualization tools in Python, such as matplotlib or seaborn, facilitates the presentation of results in a readily comprehensible format.

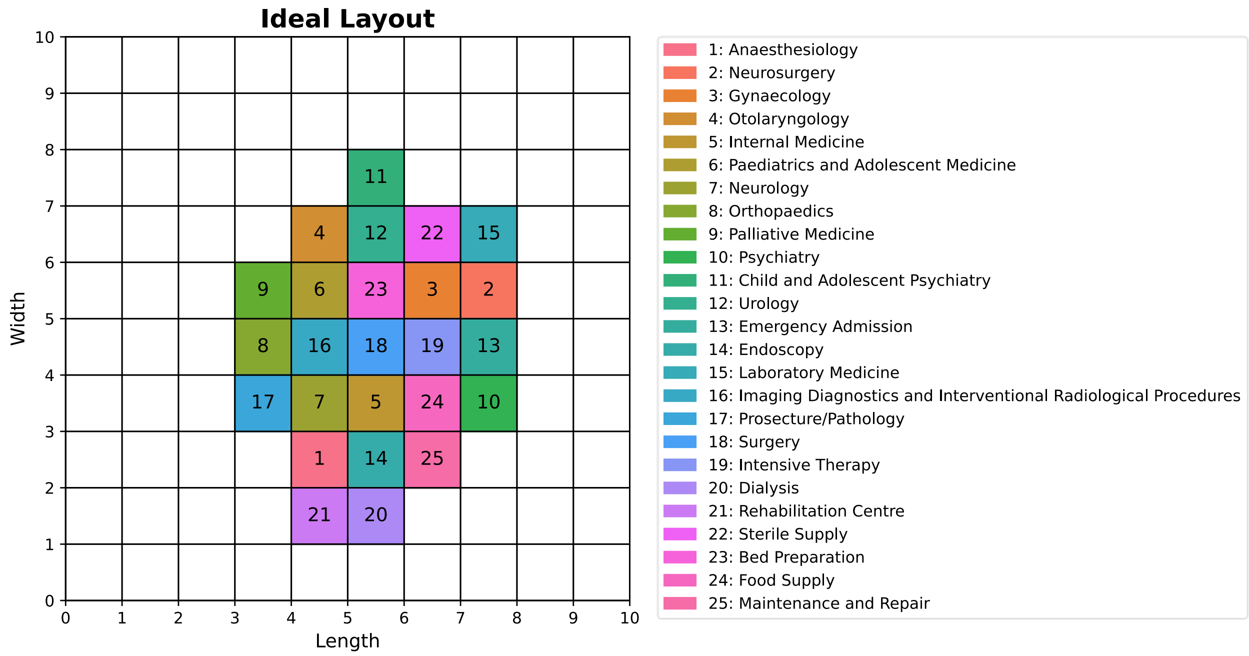

Figure 8: Example of layout visualization using the Matplotlib library in Python

Based on the initial results, perform parameter tuning in order to optimize the performance of the algorithm. This iterative process entails modifying the algorithmic hyperparameters with the objective of enhancing the quality of the results, thereby facilitating the acquisition of more accurate and meaningful insights.

5. Application and Discussion

In order to test the algorithm and the quality of the results it can produce, two function schemes are compared below, based on an abstract case study. One function scheme was created by a hospital expert and the other with the help of the algorithm. The flow data used for the case study comes from a large city hospital that provides specialized care. The hospital has around 750 beds in just over 20 departments. Due to ongoing staff shortages, the requirement for the function scheme to be created was to reduce the transport efforts for the transport staff and thus relieve the employees in the long term. The orders of the transport staff are determined by KIS, but this does not show the exact number of transport staff. The number of transport staff required for the transport and the corresponding personnel costs can therefore be determined indirectly from the orders for transport objects 1.01, 1.02 and 3.01. After cleaning the data, e.g. by excluding cancelled transport orders, the data still needs to be manipulated so that the actual transport costs can be determined and a flow matrix created from them. What cannot be seen in the data of the example given and must therefore be taken into account when weighting is that transport object 1.01 was transported by two transport staff in about 20% of cases and that objects 1.01 and 3.01 are more cumbersome to transport and, according to the estimates of the transport staff, take about 25% longer. The manipulation of the data is as follows:

$$n_{ts} = n_{1.01}\left(1 + (w_{a,1.01} + w_{te,1.01})\right) + n_{1.02}\left(1 + (w_{a,1.02} + w_{te,1.02})\right) + n_{3.01}\left(1 + (w_{a,3.01} + w_{te3.01})\right) \qquad (8)$$

with $n_{1.01}, n_{1.02}$ and $n_{3.01}$ as the frequency of transport orders per transport object recorded in the data, with the weighting factors $w_{a,1.01}, w_{a,1.02}$ and $w_{a,3.01}$ for the number of transport personnel required for the respective transport and with the weighting factors $w_{te,1.01}, w_{te,1.02}$ and $w_{te,3.01}$ for the time required for each transport object. Based on the flow data for objects 1.01, 1.02 and 3.01, the flow matrix shown in Figure 9 is derived from the above assumptions and used as input for the creation of the function schematics.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
a		0	3	7	84	0	12	21	0	0	0	16	6	2	0	44	0	60	0	20	27	0	290	0	1
b			2	0	4	0	5	2	0	0	0	0	0	0	0	2	0	11	0	0	0	0	0	0	0
c				19	446	1	45	4	7	2	1	34	184	380	0	854	10	2712	70	64	0	887	8393	2904	92
d					403	1	85	33	26	8	0	8	125	80	0	827	2	1830	20	647	0	295	2705	0	63
e						0	6183	577	201	294	0	241	1647	8011	0	14035	124	3345	59	1218	0	0	11233	0	413
f							1	2	0	0	0	0	169	0	0	13	0	5	4	0	0	0	5816	0	91
g								32	21	68	0	101	535	1145	0	11907	62	1921	240	380	0	0	9745	1	119
h									3	2	0	44	97	203	0	7368	0	870	516	88	0	0	2939	0	41
i										3	0	14	19	150	0	775	54	30	2	28	0	0	2103	1	210
j											0	9	702	90	0	931	0	39	0	28	0	0	1414	2166	293
k												0	8	0	0	5	0	0	2	2	0	0	241	0	6
l													106	87	0	1089	16	1413	176	142	0	0	4228	0	260
m														270	12	895	10	735	52	108	0	644	969	2	256
n															0	280	0	4919	94	55	0	295	66	0	27
o																0	0	0	6	0	0	0	0	0	0
p																	0	18539	1658	203	3	0	28	0	4
q																		106	122	0	0	0	0	0	0
r																			1278	1399	0	0	16771	3842	850
s																				118	0	0	2680	1920	11
t																					0	0	7	0	11
u																						0	7	0	0
v																							0	0	0
w																								0	1
x																									0
y																									

1 -	1000
1001 -	3000
3001 -	7000
7001 -	10000
10001 -	…

a: Anesthesiology; **b:** Neurosurgery; **c:** Gynecology; **d:** Otolaryngology; **e:** Internal Medicine; **f:** Pediatrics and Adolescent Medicine; **g:** Neurology; **h:** Orthopedics; **i:** Palliative Medicine; **j:** Psychiatry; **k:** Child and Adolescent Psychiatry; **l:** Urology; **m:** Emergency Admission; **n:** Endoscopy; **o:** Laboratory Medicine; **p:** Imaging Diagnostics and Interventional Radiological Procedures; **q:** Prosecture/Pathology; **r:** Surgery; **s:** Intensive Therapy; **t:** Dialysis; **u:** Rehabilitation Centre; **v:** Sterile Supply; **w:** Bed Preparation; **x:** Food Supply; **y:** Maintenance and Repair

Figure 9: Flow matrix of movement frequency between structure elements – example of patients in bed

5.1 Function Scheme by Hospital Expert

Based on the flow matrix from Figure 9, a hospital expert has created a flow-oriented function scheme. She is a general physician who has now been working as a hospital quality manager for over 20 years and is responsible, among other things, for quality audits. Since the individual flow intensities were not directly comparable for the expert from the matrix, they could not be directly related to each other. In a first step, the expert therefore created a graphical visualization. Subsequently, the area units could be arranged in relation to each other as shown in Figure 10. According to the expert, who could only make a visual assessment without further tools, this is the best arrangement to keep transport efforts as low as possible. Even though, the fitness score is 771238.75 according to Equation 7. The process was iterative, very time-consuming and challenging. Furthermore, she could only take flow intensities of 1001 transports or more into account, otherwise it would have become too complex.

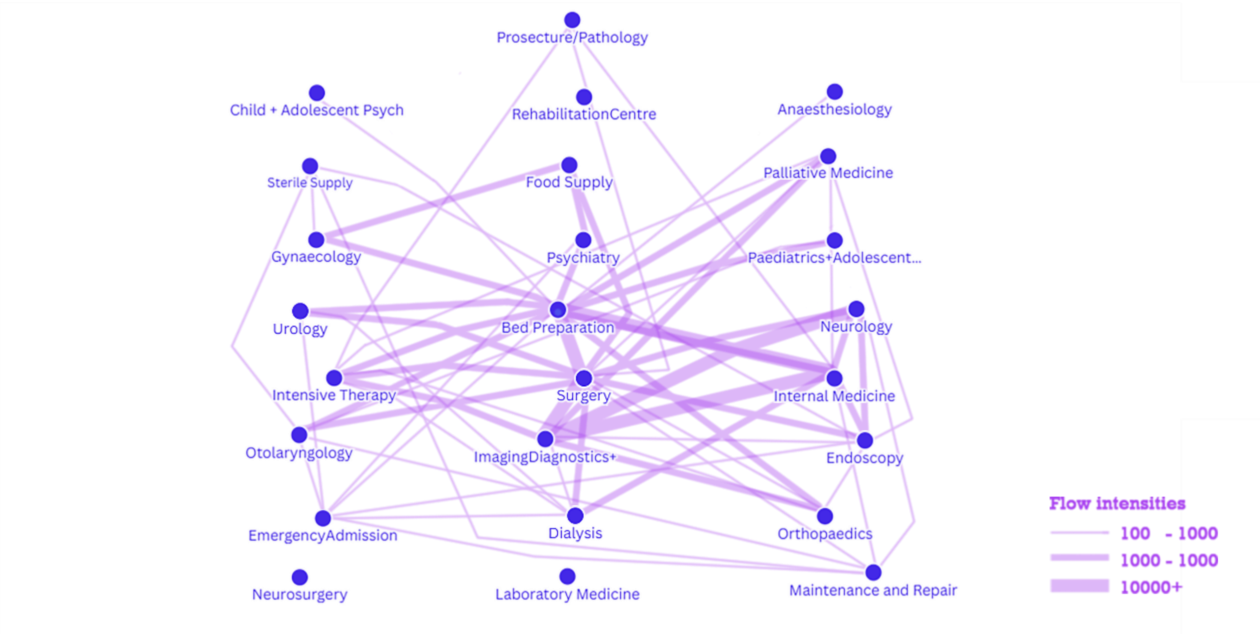

Figure 10: Function scheme created by the hospital expert (fitness score 771238.75)

5.2 Function Scheme by Algorithm

In this section, a hypothetical case study is presented. that includes 25 area units and an available 7X7 grid. The flow matrix in Table 1 shows the frequency of movements between rooms. The number of population is considered to be 50, the iteration number is 500 and w is set to 100. The crossover operation is two-point crossover and the mutation operator is reverse mutation with a probability rate of 0.0005. The penalty value is $1e9$. Figure 11 shows a graphical representation of the final solution. The achieved layout is compact, avoiding unused space between the rooms. The fitness score is 190936.00, which means the layout drawed by the algorithm is more effective than the solution provided by the expert.

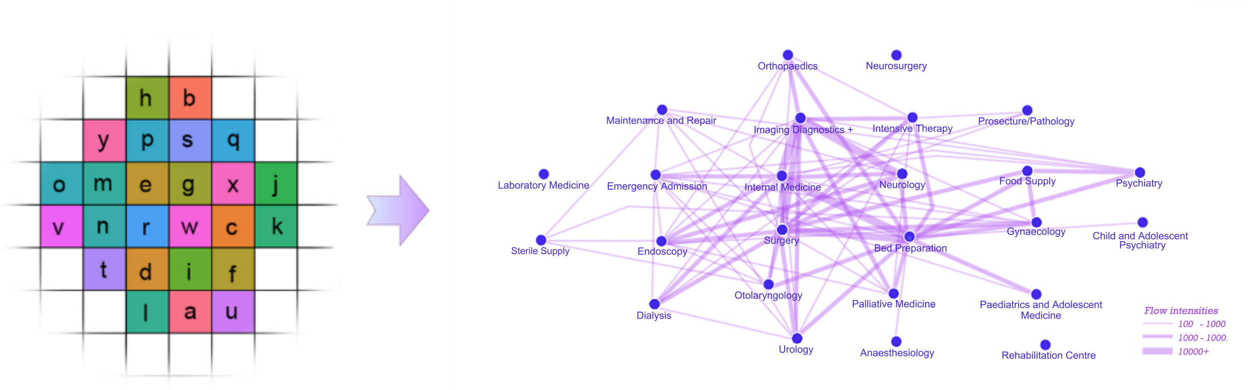

Figure 11: Function scheme created by the algorithm (fitness score 190936.00)

5.3 Discussion

Compared to the algorithm, the manual creation of the function scheme by the experts was x times more time-consuming. The manual process also required a lot of discussion. Furthermore, it was not possible for the experts to take all transport relationships into account when finding a solution and to differentiate the transport relationships more precisely than in the intensity classes shown in the figures. This exceeded the amount of information that humans can handle. The algorithm, on the other hand, was able to take into account all the transport relationships and also the exact intensities, so that the created function chart not only ensures a good solution for the areas with high transport relationships, but also for all others. The quality of the results is thus correspondingly higher thanks to the algorithm. In terms of quality and time, it makes sense to support the process of creating a function scheme with the algorithm. This can create several solution variants, which can be discussed by the experts in the project team and adapted as needed.

6. Conclusion and Outlook

For the layout planning of a hospital, the function scheme represents an important intermediate result, which can have a critical influence on the fulfilment of the target requirements such as the decrease of transportation efforts during hospital operation. The creation of the function scheme can be challenging and time-consuming due to the amount of transportation information to be processed. An approach that addresses the reduction of transport efforts with the help of corresponding layout planning and could be easily applied in practice has not existed so far, so there was no suitable support for hospital planners. The paper presents a metaheuristic algorithm based on genetic algorithms that proposes solutions for a possible function scheme based on flow matrices. In a case study, the advantage of using the algorithm to create the function scheme in terms of quality and effort was demonstrated. In addition, a detailed application guide was presented that enables practitioners to apply the approach.

Due to the limitations of the data provided by the hospital, some flow information is missing from the dataset (e.g. flow between sterile supply and anesthesia). However, the algorithm is fully adaptable to any dataset in the same format. With more complete data, the layout design will be much closer to the real needs.

In further research, the aim is to develop a procedure, including guiding principles, for deriving a realizable layout concept based on the function scheme planning approach presented in this paper. In addition, the algorithm could be extended for more specific applications, for example by formulating further restrictions that are common in hospitals.

Acknowledgements

The authors kindly thank the German Research Foundation (DFG) for funding the research project "Entwicklung von fabrikplanerischen Leitprinzipien zur Planung von flussorientierten Krankenhäusern mit wandlungsfähigen Raum-, Technik- und Organisationskonzepten (MedFAP)" (project number 456684366).

Appendix

Object Cluster		relevant
1.01	Patient in bed	
1.02	Patient in wheelchair	
1.03	Patient with NFA stretcher	
1.04	Patient with escort	
1.05	Patient deceased	
2.01	Transport staff	
2.02	Nursing staff	
2.03	Cleaning staff	
2.04	Doctors	
3.01	Bed logistics	
3.02	Sterile goods	
3.03	Catering and kitchen logistics	
3.04	Waste logistics	
3.05	Cleaning	
3.06	Pharmaceutical logistics	
3.07	Laboratory logistics	
3.08	Laundry and workwear logistics	
3.09	Business and administrative logistics	
3.10	Blood products and transplant logistics	
3.11	Medical technology logistics	
3.12	Medical product logistics	

Figure 12: Clustered objects mentioned in Chapter 4.1 for the function scheme optimization problem

References

[1] 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, 1–19.

[2] Wecken, L., Faeghi, S., Hingst, L., Nyhuis, P., Lennerts, K., 2022. General Approach and Prerequisites for Transferring Factory Planning Methods on Flow Orientation and Transformability to Hospital Systems. Proceedings of the Conference on Production Systems and Logistics: CPSL 2022, 435–444.

[3] Zuo, X., Li, B., Huang, X., Zhou, M., Cheng, C., Zhao, X., Liu, Z., 2019. Optimizing Hospital Emergency Department Layout via Multiobjective Tabu Search. IEEE Trans. Automat. Sci. Eng. 16 (3), 1137–1147.

[4] Helber, S., Böhme, D., Oucherif, F., Lagershausen, S., Kasper, S., 2016. A hierarchical facility layout planning approach for large and complex hospitals. Flex Serv Manuf J 28 (1-2), 5–29.

[5] Chraibi, A., Kharraja, S., Osman, I.H., Elbeqqali, O., 2015. Multi-Agent System for Solving Dynamic Operating Theater Facility Layout Problem. IFAC-PapersOnLine 48 (3), 1146–1151.

[6] Wecken, L., Masoudian, F., Lennerts, K., Nyhuis, P., 2024. General Guidelines For The Structural Planning Of A Hospital Following The Example Of Factory Planning Including Decision Support For The Design Of Individual Structural Elements: A German Solution Approach. Journal of Production Systems and Logistics.

[7] Wiendahl, H.-H., Reichardt, J., Nyhuis, P., 2024. Handbuch Fabrikplanung: Konzept, Gestaltung und Umsetzung wandlungsfähiger Produktionsstätten, 3., vollständig überarbeitete Auflage ed. Hanser, München, XX, 553 Seiten.

[8] Verband Deutscher Ingenieure, 2011. Fabrikplanung: Planungsvorgehen.

[9] Michalewicz, Z., Schoenauer, M., 1996. Evolutionary Algorithms for Constrained Parameter Optimization Problems. Evolutionary Computation 4 (1), 1–32.

[10] Katoch, S., Chauhan, S.S., Kumar, V., 2021. A review on genetic algorithm: past, present, and future. Multimedia tools and applications 80 (5), 8091–8126.

[11] Mirjalili, S., 2019. Genetic Algorithm, in: Mirjalili, S. (Ed.), Evolutionary Algorithms and Neural Networks, vol. 780. Springer International Publishing, Cham, pp. 43–55.
[12] Syswerda, G., 1989. Uniform Crossover in Genetic Algorithms. Proc. 3rd Intl Conference on Genetic Algorithms.
[13] Ahn, C.W., Ramakrishna, R.S., 2003. Elitism-based compact genetic algorithms. IEEE Trans. Evol. Computat. 7 (4), 367–385.
[14] Loiola, E.M., Abreu, N.M.M. de, Boaventura-Netto, P.O., Hahn, P., Querido, T., 2007. A survey for the quadratic assignment problem. European Journal of Operational Research 176 (2), 657–690.
[15] Deutsches Institut für Normung e. V., 2016. Division of hospitals into functional areas and functional sections.

Biography

Yuhao Gao, M.Sc. (*1993) studied mechatronic with a focus on robotic technology at Karlsruhe Institute of Technology (KIT). He works as a research associate in Facility Management group at Institute of Technology and Management in Construction (TMB) at KIT since 2023, where he focuses on applying AI technology in hospitals.

Lena Wecken, M.Sc. (*1995) studied industrial engineering with a focus on production technology at Leibniz University Hannover and Athens University of Economics and Business. She works as a research associate in the specialist factory planning group at the Institute of Production Systems and Logistics (IFA) at the Leibniz University Hannover since 2021.

Prof. Dr.-Ing. Kunibert Lennerts (*1963) is holding the professorship for Facility Management at Karlsruhe Institute of Technology since 2000 and was appointed to the first university professorship in Facility Management in Germany. His main research areas are facility management and sustainability with special focus on huge real estate portfolios and hospitals.

Prof. Dr.-Ing. habil. Peter Nyhuis (*1957) studied mechanical engineering at Leibniz University Hannover and subsequently worked as a research associate at IFA. After completing his doctorate in engineering, he received his habilitation before working as a manager in the field of supply chain management in the electronics and mechanical engineering industry. He is heading the IFA since 2003.