Outils d'aide à la décision pour la conception des blocs opératoires

Abdelahad Chraibi

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITE JEAN MONNET DE SAINT ETIENNE

Spécialité : Informatique Industrielle

Délivré par :

Université Jean Monnet de Saint Etienne

Présentée par :

CHRAIBI Abdelahad

Le 10 Décembre 2015

Titre :

A Decision Making System for Operating Theater Design: Application of Facility Layout Problem

Ecole Doctorale Sciences, Ingénierie, Santé - ED SIS 488

Unité de recherche :

Laboratoire d'Analyse des Signaux et des Processus Industriels – LASPI – EA 3059

Jury:

Said HANAFI	Professeur, Université de Valenciennes, France	Rapporteur
Farouk YALAOUI	Professeur, Université de Troyes, France	Rapporteur
Laetitia JOURDAN	Professeur, Université de Lille, France	Examinatrice
Christophe CORBIER	Mcf, Université Jean Monnet de Saint Etienne, France	Examinateur
François GUILLET	Professeur, Université Jean Monnet de Saint Etienne, France	Membre invité
Ibrahim Hassan OSMAN	Professeur, Université Américaine de Beyrouth, Liban	Directeur de Thèse
Omar ELBEQQALI	Professeur, Université Sidi Mohammed ben Abdellah, Fès, Maroc	Directeur de Thèse
Said KHARRAJA	Mcf, Université Jean Monnet de Saint Etienne, France	Co-Directeur de Thèse

A Decision Making System for Operating Theater Design: Application of Facility Layout Problem

Abstract

In the last decades, the important increasing consumption of health care and the growing of population make elimination of waste and continuous productivity improvement more and more critical for hospitals to provide their care services effectively and efficiently. The productivity and efficiency of a hospital depends on the caregivers working conditions, which are impacted greatly by the work place and the facilities organization [Dares (2013)]. Facilities planning "determines the physical organization of a production system and finding the most efficient arrangement of 'n' indivisible facilities in 'n' locations" [Singh & Sharma (2006)]. Thus, facilities planning has a great impact on the productivity and efficiency of running a hospital.

Being aware of this need, the work we present aims to find a solution to facilities planning for the Operating Theater "the heart of hospital" by proposing an intelligent tool we make available to decision makers for optimizing their operating theater design. Our research work focuses on the use of operational research methods in order to find a solution for this optimization problem. Methods we explored for the realization of this work were variant, namely exact algorithm, heuristics, metaheuristics and intelligent methods, which allow us to compare different issues in order to provide the best solution to different scenarios of problems.

Thus, in this dissertation we present the major contribution of our work, starting with the application of Mixed Integer Programming (MIP) to solve Operating Theater Layout Problem (OTLP) as the first scientific contribution. This work considers three different formulations (i.e. the multi-sections, the multi-floors and the multi-rows) in two different environment types (i.e. static and dynamic) while optimizing two different objective functions (i.e. to minimize the total traveling cost and to maximize the total adjacency rate). The combination of these different components gives rise to nine MIP models to solve the OTLP for which optimal solution was provided to problems with until forty facilities. These contributions are presented in the third and fourth chapters.

The use of Multi-Agent System (MAS) to solve Facility Layout Problem (FLP) is the second scientific contribution we present in chapter five. In literature, only one work [Tarkesh et al., (2009)] applied the MAS to solve small sized problems, which makes our work the first one adopting MAS to address both the static and dynamic FLP for large sized problems using a novel algorithm running in three steps to solve OTLP. The developed multi-agent platform exploit the three different agents' protocols of communication, namely coordination, cooperation and negotiation to conceive different agents' architectures to deal with the static and dynamic OTLP.

The last contribution consisting on the use of Particle Swarm Optimization (PSO) under continuous layout representation to solve multi-rows FLP is presented in chapter six. Since the PSO is generally used to solve assignment problems or discrete FLP, the actual formulation is among the few works dealing with the continuous one. This leads us to conceive a novel encoding technique and the appropriate heuristics to generate initial solutions and to perform the local search procedure. Another novelty is related to the application of PSO to a multi-rows layout problem, which was not addressed before. To the best of our knowledge, PSO works usually formulate the FLP as a single row or in the best of scenarios, as a double-rows problem.

Résumé

Dans les dernières décennies, l'importante augmentation de la consommation des services de soins et la croissance de la population on fait de l'élimination du gaspillage et l'amélioration continue de la productivité de plus en plus cruciale pour les hôpitaux. La productivité et l'efficacité d'un hôpital dépendent des conditions de travail des soignants qui sont influencés fortement par l'organisation des lieux de travail et des installations [Dares (2013)]. L'agencement des installations consiste à "déterminer l'organisation physique d'un système de production et de trouver l'arrangement le plus efficace de 'n' installations dans 'n' positions" [Singh et Sharma (2006)]. Ainsi, l'agencement des installations a un grand impact sur la productivité et l'efficacité du fonctionnement d'un hôpital.

Etant conscient de ce besoin, le travail que nous présentons vise à trouver une solution à l'agencement des salles du Bloc Opératoire "le cœur de l'hôpital", ainsi que les salles annexes, à savoir, salle d'induction, salle de réveil, etc… en proposant un outil intelligent que nous mettant à la disposition des maitres d'ouvrages pour optimiser leur conception du bloc opératoire. Notre travail de recherche se concentre sur l'utilisation des outils de la recherche opérationnelle afin de trouver une solution à ce problème d'optimisation. Les méthodes que nous avons explorées pour la réalisation de ce travail sont variantes, à savoir les méthodes exactes, les heuristiques, les métaheuristiques et les méthodes intelligentes, ce qui nous a permis de comparer les différentes approches afin de fournir la meilleure solution pour différents scénarios de problèmes.

Ainsi, dans cette thèse, nous présentons les contributions majeures de notre travail, à commencer par l'application de la programmation mathématique en nombres entiers mixtes (Mixed Integer Programming (MIP)) pour résoudre le problème d'agencement du bloc opératoire (Operating Theater Layout Problem (OTLP)) comme la première contribution scientifique. Ce travail considère trois structures différentes (multi-section, multi-étage et multi-rangé) dans deux types d'environnement différents (statique et dynamique), tout en optimisant deux fonctions objectifs différents (minimiser le coût de déplacement totale et de maximiser le taux totale de proximité entre les différentes salles). La combinaison de ces différentes composantes donne lieu à neuf modèles MIP pour résoudre l'OTLP pour lesquels une solution optimale a été atteinte pour des problèmes avec jusqu'à quarante salles. Ces contributions sont présentées dans le troisième et quatrième chapitre.

L'utilisation de Systèmes Multi-Agents (MAS) pour résoudre le problème d'agencement des installations (Facility Layout Problem (FLP)) est la deuxième contribution scientifique que

nous présentons dans le cinquième chapitre. Dans la littérature, on retrouve un seul travail [Tarkesh et al., (2009)] ayant appliqué le MAS pour résoudre des problèmes de petites tailles, ce qui rend notre travail, le premier adoptant MAS pour répondre à la fois le FLP sous environnement statique et dynamique pour des problèmes de grande taille en utilisant un algorithme en trois étapes pour résoudre OTLP. La plate-forme multi-agents développée exploite les trois différents protocoles de communication d'agents, à savoir la coordination, la coopération et la négociation pour concevoir différentes architectures d'agents afin de faire face à l'OTLP statique et dynamique.

La dernière contribution consistant en l'utilisation de l'optimisation par essaim de particules (Particle Swarm Optimization (PSO)) sous une représentation continue de l'espace de recherche pour résoudre le problème d'agencement multi-rangée est présentée dans le sixième chapitre. Puisque la PSO est généralement utilisé pour résoudre les problèmes d'affectation ou les FLP avec une représentation discrète, la formulation actuelle est parmi les rares travaux traitant la représentation continue du FLP. Pour y parvenir, nous avons conçu une nouvelle technique de codage des particules et des heuristiques appropriées pour générer des solutions initiales et pour effectuer la procédure de recherche locale. Une autre nouveauté est liée à l'application de la PSO à un problème de structure multi-rangé, qui n'a pas été abordé auparavant car à notre connaissance, les travaux avec la PSO ont formulé le FLP comme une structure d'une seule rangée ou dans le meilleur des scénarios, comme une structure à deux rangées.

Acknowledgment

All thanks are to God first for giving me the courage and strength I needed to achieve this work. I want then to thank my three supervisors M Said Kharraja, M Ibrahim Hassan Osman and M Omar El Beqqali for their constant encouragement and guidance, for their friendly attitude and for the quality of their supervision in my three years of doctoral study. The completion of this thesis would not have been possible without their support and advisory. I learned so much from them as a model of teaching and research excellence.

I feel grateful to all the heads of the LASPI laboratory, especially M François Guillet and M Mohamed El Badaoui for welcoming me to the laboratory and giving me the chance to work in a pleasant atmosphere. I would like to express my heartfelt gratitude to M Nabih Nejjar the director of the IUT of Roanne, M Salah Khennouf the director of the CUR of Roanne and M Thierry Delot the director of the ISTV of Valenciennes for giving me the chance to teach in their institutions and develop my personal and professional skills.

I am honored and appreciative to the participation of the thesis committee: M Said Hanafi, M Farouk Yalaoui, M François Guillet and M Christophe Corbier. Thanks to you for accepting to take part in the jury and to judge my work.

I had the great opportunity to benefit from the mobility grant financed by Erasmus Mundus 'Al-Idrisi'. Without your financial support, this thesis could not be conducted. I want to say thank you for providing us this chance, thanks for M Thomas Guillobez for his coordination, his support and for having facilitated all the procedures.

I would like to tanks all my LASPI colleagues Malek Masmoudi, Sofiane Maiz, Mourad Lamraoui, Amine Brahmi, Firas Zakaria, Mohammed Telidjane, Thameur Kidar, Rabeh Redjem, Donald Rotimbo, Ramzi Halabi, Claude Vivien Toulouse, Marouane Frini, Nezha Zeriouh, Michelle Billet, Christophe Corbier, Lucien Perisse, Ikram Khatrouch, Sylvain Pamphile Tsangou …. I will always have fond memories of these years spent with you. You have continued to encourage me and support me in my work and in difficult times. I thank you for encouragement and I want to tell you that your friendship was and still really important to me.

Dr Ahmed Kadour Belgaid, M Mazouni Hadj Abdelkader, Salim Arezki, Farid Azili, Khalid Ayadi, Jamal Benmousah, Abdelah Toumi, Yassine Mechtaoui, Ali Mechtaoui, Abdelah Choitel, Yassine Badri, Ounais Madi... thanks to all of you my friends for being here. Your presence always lights up my way and gives me inspiration and courage to continue the adventure.

My great thanks go to my father Abderrafih Chraibi. All I have and will accomplish are only possible due to your love, support and sacrifices. My brother Abdelali and sisters Asmae and Khawla, I want to express my gratitude to you for your inspiration and encouragement. Thanks to my uncles, my aunts, my cousins, and my wife's family for their support.

They said 'Behind every great man there is a great woman', but I had two great women to whom I dedicate this success. Great mother, without you this work could not been done. You were the source of my energy and the light of my life. Your prayers for me have made everything I have done possible. Dear wife Hind, I express my love and my gratitude to you for your loving considerations and great confidence in me all through these years. Your patient and support was instrumental in accomplishing this work.

Contents

Sommaire

List of Tables

List of Figures

Chapter I.

Introduction

The management of an Operating Theater (OT) in hospitals has been proven to be a complex process. First, patients who need to be admitted to hospitals have to be convinced of the quality and safety of the operation of this mysterious and symbolic place, especially in a world competitive health-care market. Second, the professional body also working in such places require an optimal architectural and technical environment to ensure the effectiveness and efficiency of their operating interventions. Finally, the management teams are under increasing pressures to meet new quality standards imposed by regulatory and accreditation boards. All these factors make the design of OT layout a complex project and of prime importance to assure satisfaction of all stakeholders to facilitate the movement of both patients and health-care providers.

The research outlined in this dissertation aims to provide modeling and solving methodologies to help hospitals' management with analytical tools to improve the efficiency and effectiveness of their operating theater layout. In Section 1.1, we provide motivation to conduct such research project. In Section 1.2, we provide the background and best-known approaches for the facility layout problem (FLP) and the commonly used objective functions. In Section 1.3, our research objectives are provided. Finally, in Section 1.4 we present the outline of this dissertation.

1.1. Motivation

The current political, economic and social context in France, characterized by seeking to streamline costs, control health care outgoings, the increasing consumption of health care and the aging population, encourages hospitals to rethink their conception and restructure their organizations in order to provide high quality care services at lower cost while maintaining the comfort and wellbeing of staff.

The importance of hospital's design has been proven to have a great impact on the hospital's safety. For instance, a report by the Joint Commission on Accreditation of Healthcare Organizations cited the physical environment as a root cause for 50% of patient falls. In a survey on Working Conditions conducted by the Minister of Labor in France, reported that more than 67% of employees are not able to do their jobs properly due to inappropriate local facilities (Dares [2013]).

In addition, some statistics found that 28.9% of nursing time was spent on walking. Further, a poorly designed physical environment creates latent conditions such as staff stress, accident, and retarding patients that may potentially lead to adverse events in hospitals and affect the quality of care services.

The OT in a hospital holds a prominent place in a hospital's performance. In fact, the capacity of hospital and its attractive image are based to a large extent on the reputation of the OT outcomes. The OT expenses are increasingly one of the most services in a hospital both in terms of investment budget and in terms of functionality. The OT has also a significant impact on performance of other hospital' services. In fact, it plays a bit an organizational pivotal role in the hospital.

Many research works related to OT are present in the literature. They can be grouped in the six principal fields and are illustrated in Figure 1.1. Although all fields are interdependent, our OT work addresses the third field, it deals with the implementation and disposition, which seeks to determine the 'optimal placement' of a set of facilities within the available space subject to constraints imposed by the site plan, the building, the departmental area, the service requirements, and the decision-makers. The importance of this topic came from the great impact

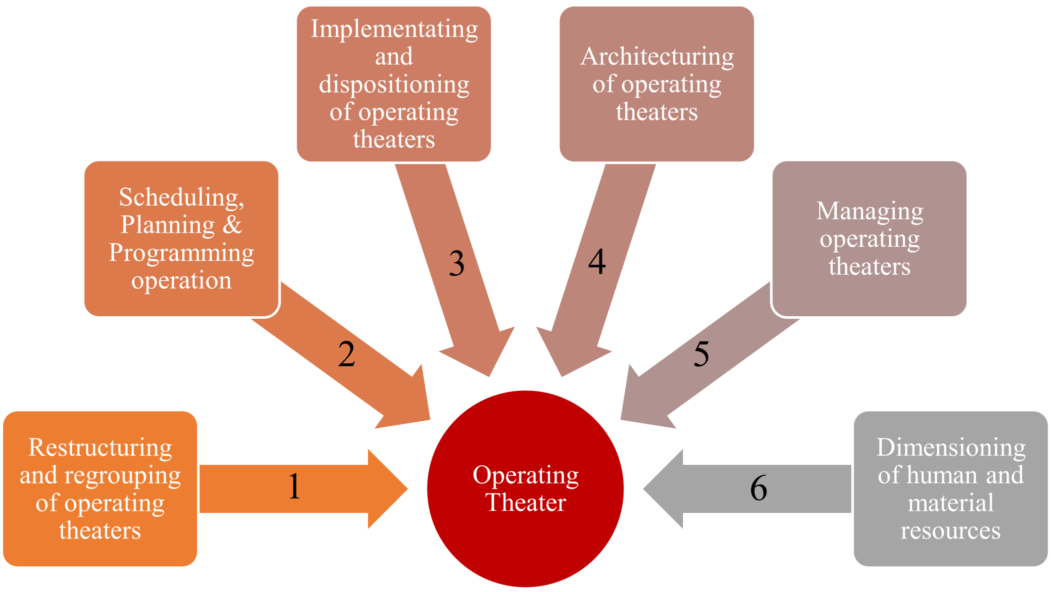


Figure 1.1. Research fields related to OT

of the OT design on its operating functions, the safety and satisfaction of patients and the wellbeing of health-care providers.

Unfortunately, for several decades, the design and establishment of OT has been realized by architects using traditional specifications based on their experiences, design aspects, and certain predefined standards. They do not integrate intelligent decision-making systems. As a result, the final OT design may not be optimized to deal with all the flows of patients, materials, medical and non-medical staff.

In light of these shortcomings, providing an innovative tool for the OT layout design to consider hospital objectives, optimization aspects, and international standards on regulation and architecture has become an ambitious challenge to be addressed in this work.

We aim to develop a decision-making system, which allows decision makers to design an efficient OT by introducing modifications to the Facility Layout Problems (FLP). Thus, the OT layout has been generated using mixed integer linear programming (MILP) models according to the environment representation {e.g. statistic FLP (SFLP) and dynamic FLP (DFLP)} and different design configurations (e.g. multi-section layout, multi-floor layout and multi-row layout) to design a simple OT structure for small-sized hospitals.

For larger hospitals where the structure of OT is more complex, we introduced the use of Multi-Agents Systems (MAS) and Particle Swarm Optimization (PSO) for the different environment representations and design configurations.

1.2. Facility Layout Problem

1.2.1. Background

The Facility Layout Problem can be defined as determining the physical organization of a production system and the arrangement of everything needed for the production of goods or the delivery of services. FLP has wide applications in the design of hospitals, assembly lines, airports, warehouses, offices, among others. It is known to have a great impact on the productivity, manufacturing costs, lead times and efficiency of running its adopted organization. A good placement of facilities contributes to the overall efficiency of operations and it can reduce the total operating expenses by up to 50% (Tompkins et al., [2010]).

Many research works have been published in this field. The FLP has been defined as follows:

"*The FLP is concerned with finding the most efficient arrangement of m indivisible departments with unequal area requirements within a facility*." Meller and Gau (1996)

"*The FLP is an arrangement of everything needed for production of goods or delivery of services. A facility is an entity that facilitates the performance of any job. It may be a machine tool, a work center, a manufacturing cell, a machine shop, a department, a warehouse, etc*." Heragu (1997)

"*The FLP is a family of design problems involving the partition of a planar region into departments or work areas of known area, so as to minimize the cost associated with projected interactions between these departments*." Shouman et al. (2001)

"*The FLP* consists *in arranging n unequal-area facilities of different sizes within a given total space, which can be bounded to the length or width of site area in a way to minimize the total material handling cost and slack area cost.*" Lee and Lee (2002)

"*The FLP is an optimization problem that tries to make layouts more efficient by taking into account various interactions between facilities and material handling systems while designing layouts*." Shayan and Chittilappilly (2004)

"*The FLP* consists *in determining the physical organization of a production system and finding the most efficient arrangement of 'n' indivisible facilities in 'n' locations*." Singh & Sharma (2006)

From these definitions and others present in the literature, the FLP is concerned with finding the optimal or best arrangement of n facilities within the available space subject to a set of constraints while optimizing a given set of objectives.

1.2.2. Layout classifications

The FLP can be classified according to different aspect described in (Drira et al. [2007]) and resumed in Figure 1.2 namely, the layout evolution, the layout representation, the manufacturing systems, the layout configuration, the facilities shapes, the layout objectives, the layout constraints, etc.

i) The layout evolution

The FLP can be classified into two main categories according to material handling flows, (1) Static Facility Layout Problem (SFLP) and (2) Dynamic Facility Layout Problem (DFLP). The SFLP assumes that the flows between facilities and demands of products are constant in a single time period. On the Contrary, the DFLP is concerned with the evaluation and modification of layouts over multiple periods; it extends the SFLP by assuming that the material handling flows and product demands can change over time. In DFLP it is necessary to periodically evaluate the changes in product demands to determine the need for rearranging the layout, in order to maintain a good facility layout. (Drira et al., [2007])

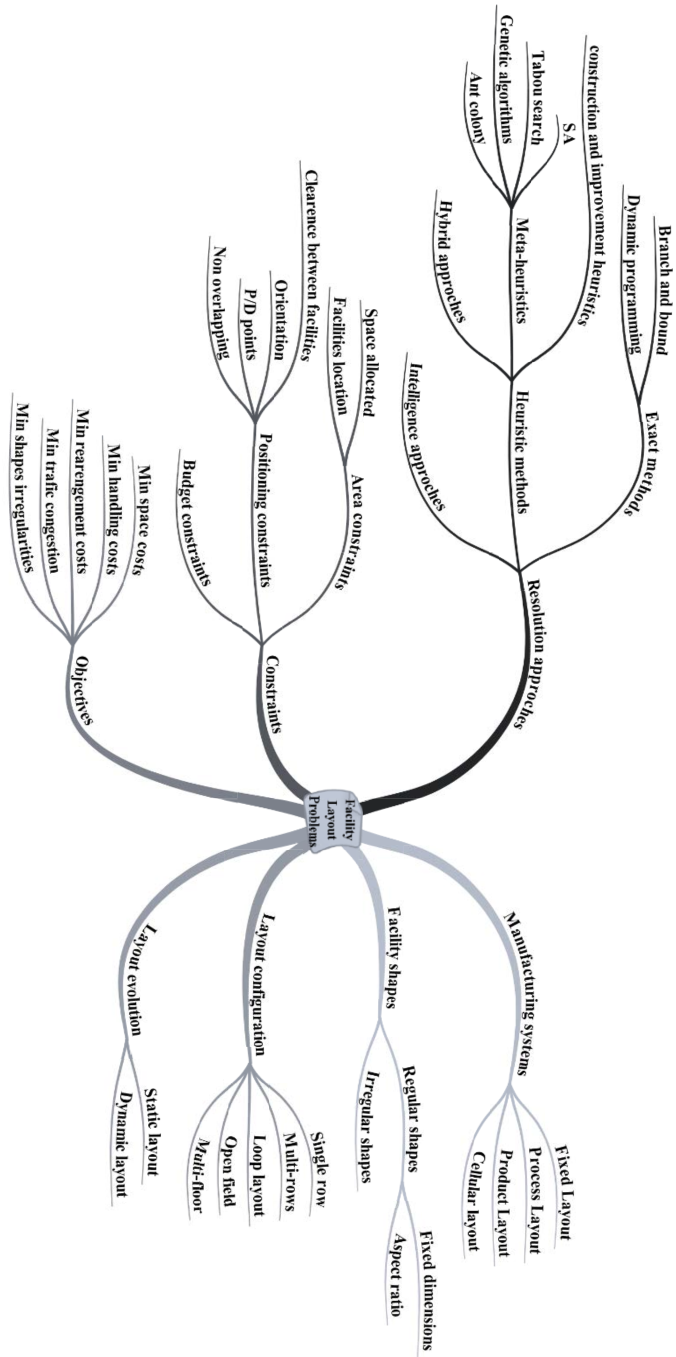


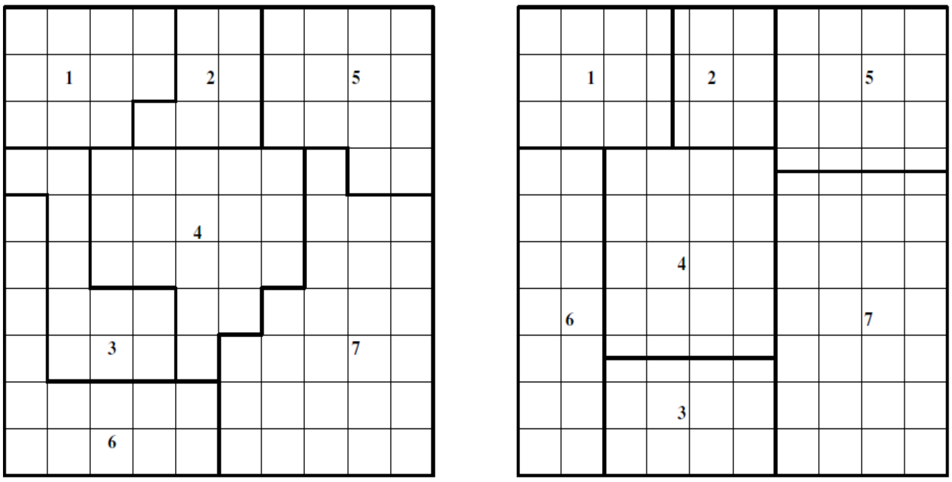
Figure 1.2. Tree representation of the layout problems classification

ii) The layout representation

The FLP can also be classified depending on the representation method. Generally, there are two representations (Tompkins et al., [2010]):

- *Discrete Representation:* the facility is represented by an underlying grid structure, (as shown in Figure 1.3.a) where each department is rounded off to an integer number of grids;

- *Continuous Representation:* department's dimensions are not restricted to an underlying grid structure, but rather, as in Figure 1.3.b, dimensions can take on non-integer values.



(a) **(b)**

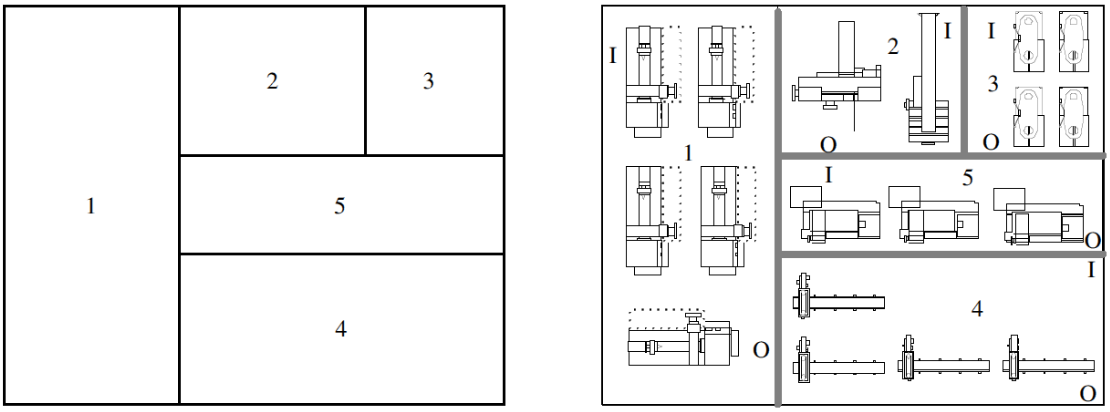
Figure 1.3. Layout with (a) Discrete Representation and (b) Continuous Representation

We can say that the continuous representation is more precise and representative than the discrete representation. The continuous representation increases the complexity of solving the FLP; however, it is able to find the "real optimal" final layout solution.

iii) The design level

Layout types are based on the design levels. According to Tompkins et al (2010), there are two levels:

- *The block layout*: is concerned with the macro flows in the facility (the design shows the



(a) **(b)**

Figure 1.4. Layout with (a) Discrete Representation and (b) Continuous Representation

location and size of each department as in Figure 1.4.a)

- *The detailed layout*: is concerned with the micro flows (the design shows the exact locations of each facility, aisle structures, input/output (I/O) point locations, and the layout within each facility, see Figure 1.4.b).

iv) Manufacturing system

Layout types are based on the material flow system. According to Tompkins et al (2010), there are four types as follows:

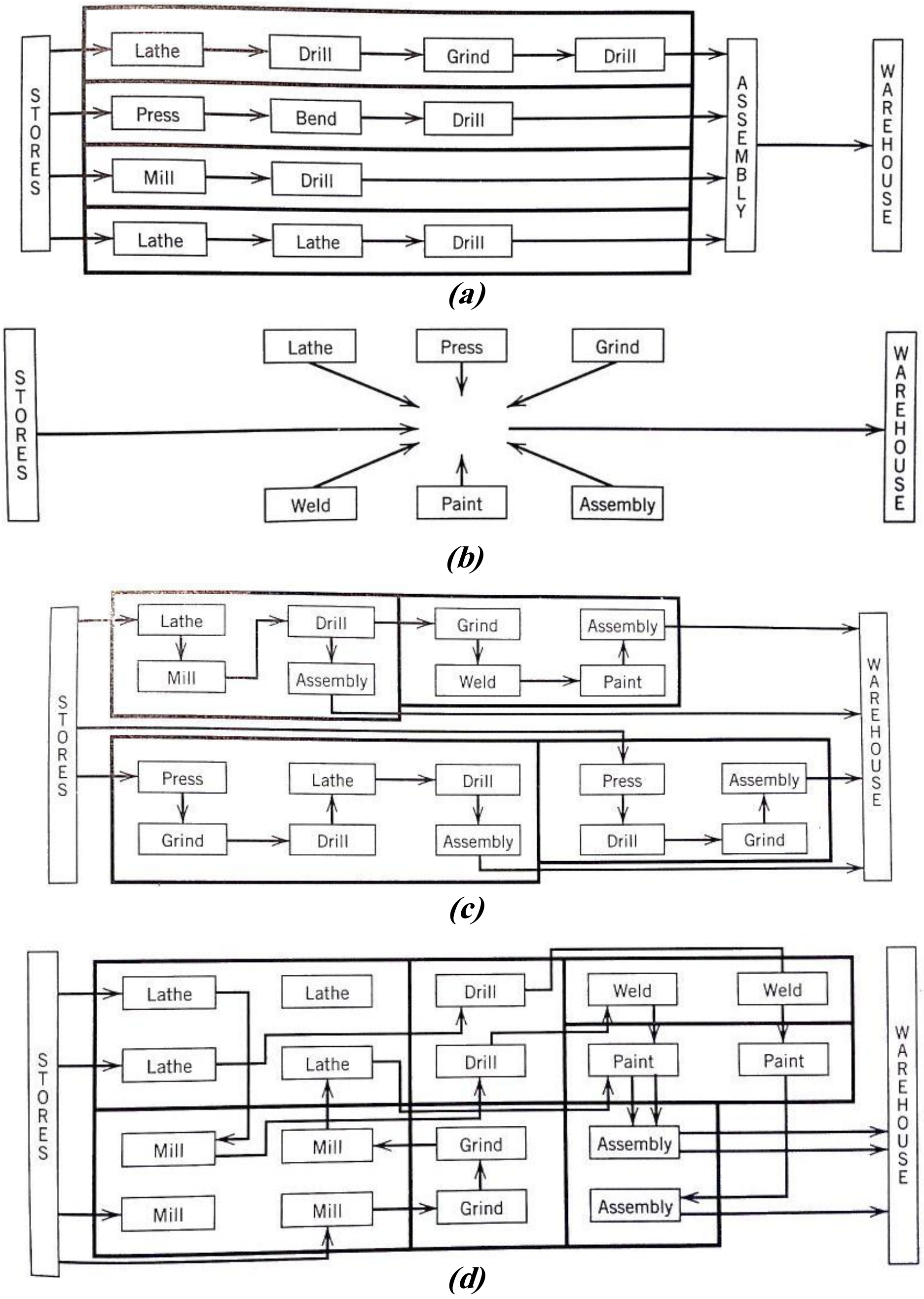


Figure 1.5. Layout types according to manufacturing system

- *Production line layout:* this type is based on the processing sequence of the parts being produced on the line. The flow of products moves directly from one workstation to the next adjacent workstation in the same line and there are often multiple lines. (Figure 1.5.a)

- *Fixed product layout:* in this particular type of layout, the product does not move, it involves displacement of workstations around the product to perform the operations on it. (Figure 1.5.b)
- *Production family layout:* this type is based on grouping of workstations into entities to form product families. Workstations can be grouped into families according to common processing sequences, tooling requirements, handling/storage/control requirements etc. (Figure 1.5.c)
- *Process layout:* in this type, the layout for a process department is obtained by grouping same processes together and placing individual process departments relative to one another based on the flow between departments. (Figure 1.5.d)

v) Objectives

The FLP aims to find an efficient non-overlapping arrangement of n facilities within a given space. In the FLP literature, the minimization of the total traveling cost and maximization of the total closeness rating between each two departments are often the two common objectives used (Meller & Gau [1996]).

- *Minimizing the total traveling cost function*: we define the distance-based function as follow:

$$\mathbf{Min}\sum_{i}\sum_{i}(f_{ij}c_{ij})\,d_{ij}$$

Where f_{ij} is the material flow from facility i to facility j, c_{ij} is the unit cost (i.e. the cost to move one unit of load from facility i to facility j), and d_{ij} is the distance from facility i to facility j.

The distance between the centroid coordinates of two facilities can be expressed by the rectilinear form:

- For layout with only one floor:

$$d_{ij} = |x_i - x_j| + |y_i - y_j|$$

- For a multi-floor layout:

$$d_{ij} = |x_i - x_e| + |y_i - y_e| + \varepsilon|F_i - F_j| + |x_e - x_j| + |y_e - y_j|$$

Where: $(x_i, y_i); (x_j, y_j); (x_e, y_e)$ are respectively the coordinates of the first facility, second facility and the elevator from the (0,0) point.

F_i = Floor (i) on which first facility is located.

F_j= Floor (j) on which the second facility is located.

ε = Factor for vertical direction transport and waiting times.

The distance can be expressed considering to the pick-up and drop-off points (input and output points). In this case, the distance traveled is calculated from the drop-off of facility i to the pick-up of facility j:

$$d_{ij} = |x_i^O - x_j^I| + |y_i^O - y_j^I|$$

Where: (x_i^I, y_i^I); (x_i^O, y_i^O) are respectively the coordinate of the input and output point for facility i.

- *Maximizing the closeness rating function*: which can be expressed by:

$$\textbf{Max} \sum_i \sum_i (r_{ij})\, x_{ij}$$

Where r_{ij} represent the numerical value of a closeness rating between facilities i and j, and x_{ij} equals 1 if facilities i and j are adjacent, and 0 otherwise based on the distance separating them.

- *The weighted sum function*:

Knowing the advantages and disadvantages of the two objectives (Meller & Gau [1996]), some authors combined the two objectives in a weighted form as follow:

$$\textbf{Min}\ \alpha \sum_i \sum_i (f_{ij} c_{ij})\, d_{ij} - (1 - \alpha) \sum_i \sum_i (r_{ij})\, x_{ij}$$

Where α is weights of objective functions, Meller & Gau (1996) studied how to set α to optimize the weighted criteria function.

vi) Resolution approaches

Any type of FLP, whether it is with SFLP/DFLP or Discrete/Continuous formulation is solved using two three of resolution approaches:

- *Exact methods:*

In theory, exact algorithms are used to obtain a global optimum solution to the FLP to guaranty the optimality of the final layout. Unfortunately, they are extremely time-consuming and can only consider problems with very small sizes (Osman et al. [2002]) which are far from the reality in industrial problems (30–40 departments). The well-known methods for such types are those used to solve the quadratic assignment problem, the graph based methods and the mixed integer programming models.

- Quadratic Assignment Problem (QAP)

The Quadratic Assignment Problem is a special case of the FLP. It assumes fixed and known locations, equal areas for each department and one-to-one matching between departments and locations. A typical QAP formulation is given as follows according to Koopmans & Beckmann (1957):

$$\min \sum_{i}^{n} \sum_{j}^{n} \sum_{k}^{n} \sum_{l}^{n} c_{ijkl} x_{ik} x_{jl}$$

Subject to:

$$\sum_{i}^{n} x_{ik} = 1 \qquad k = 1, \dots, n$$

$$\sum_{j}^{n} x_{jl} = 1 \qquad l = 1, \dots, n$$

$$x_{ik} \begin{cases} 1 & \text{if department i is assigned to location k} \\ 0 & \text{otherwise} \end{cases} \quad i, k = 1, \dots, n$$

Where c_{ijkl} is the cost incurred by affecting facility i to location k and facility j to location l.

Exact algorithms for solving QAP include approaches based on branch and bound, cutting planes and dynamic programming. The branch and bound algorithms are the most successful, but they are still unable to solve large-sized instances.

- Graph-Theoretic

In graph-theoretic approaches, it is assumed that the closeness ratings between facilities are known. Each facility is then represented by a node in a graph and the adjacency factors are represented by an arc connecting two adjacent facilities. The objective function in this case is to maximize the total closeness rating function.

To develop a layout using graph-theoretic approaches, we first create an adjacency graph from the given facility relationships, second we transform it to a dual graph and finally generating a block layout from the dual graph. As in QAP, optimality cannot be guaranteed for large sized problems. As result, many construction heuristics based on graph theoretic models are developed. Hassan & Hogg (1987) presented a thorough review of such heuristics.

- Mixed-Integer Programming

Compared to other methods, Mixed-Integer Programming (MIP) is a new approach originally presented in Montreuil (1990) to solve the FLP. This approach uses a distance-based objective and assumes a continuous representation of a layout. The MIP can solve problems with equal and unequal areas by specifying the location and the orientation of departments with the use of binary integer variables to prevent overlapping departments.

- *Heuristic Procedures:*

Taking into consideration the deficit of QAP, graph theory models, or the MIP to guarantee optimal solutions for solving large sized layout problems, researchers made efforts to find approximate solutions by implementing heuristic approaches. Heuristics can be classified into two categories: construction heuristics and improvement heuristics. Construction heuristics start selecting and locating successively a new department until the layout is completed to obtain a single solution, while improvement heuristics are based on improving an initial layout using improvement algorithm until no more improvements can be found using pair-wise and other type of exchanges. Those categories use either Adjacency-based or Distance-based approaches.

- *Metaheuristics:*

Osman (1995) defined metaheuristics as “an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions”. They are the most recent class of approximate methods that guide the search process of classical construction and improvement methods to escape local optimum. They are widely used to solve complex problems in different industrial fields, Osman and Kelly (1996). Meta-heuristics include tabu search (TS), genetic algorithm (GA), ant colony optimization (ACO), simulated annealing (SA), particle swarm optimization (PSO), Hyper-heuristics, among others. A recent survey of works on metaheuristics applied to FLP is provided in Barsegar (2011) while the origin of meta-heuristics is provided in Osman (1995).

1.3. Research objectives

Although there are a significant number of research works to solve the FLP by considering different layouts in various application fields, research on healthcare structure in general and OT in particular is still limited despite the importance of such environment. This work aims to fill in this gap and to propose different approaches to solve the FLP applied to OT layout problems.

The objectives of our research can be described as follow:

Our main research objective is to develop an intelligent decision-making system for hospitals to offer managers an easy, a comprehensive, a reliable and a financially viable tool to design their OT taking into account:

- the needs and constraints of surgeons and anesthetists teams;
- the availability of nurses;
- the relationship between different OT rooms and services;
- the procurement procedures of various materials;
- the stretcher technique facility;
- The recovery rooms resources.

The reliability of the developed tool is assured by following the international standards governing the design of healthcare facilities. Some of the references detailing these standards are presented here:

- AIA (American Institution of Architecture), Guidelines for Design and Construction of Health Care Facilities;
- FGI (Facility guideline institution), Guidelines for Design and Construction of Health Care Facilities;
- JCAHO (Joint Commission on Accreditation of Healthcare Organizations) Planning, Design, Construction of healthcare facilities;
- ICRA (Infection Control Risk Assessment), Matrix of Precautions for Construction & Renovation;
- ASHRAE (American Society of Heating, Refrigerating, and Air-Conditioning Engineering), ventilation standard for healthcare facility; etc.

Whereas the profitability aspect is obtained using optimization algorithms, which are run in the background to provide optimal for small-sized instances or near optimal for the OT design of larger sized instances under different specifications such as:

- The mode of planned hospitalization (classic or ambulatory);
- The number of interventions, their types, and their average durations;
- The number of operating rooms and pre-anesthesia;
- The number of people working in a block: surgeons, anesthetists, doctors and nurses; etc.

The second fixed objective of this research work is to provide exact methods to solve to optimality three different formulations under static and dynamic care demand. The three formulations consider continuous representation-based FLP and represent the way decision makers see the final OT either as a multi-section in one floor, as a multi-floors with the use of elevators or as a multi-rows using corridors to route different part of the OT.

The third objective is to solve real life problem sizes for OT having more than thirty facilities while insuring the effectiveness of the provided solution. This objective is reached using two approximate approaches namely a multi agent system and a particle swam optimization algorithm. These two methods are derived from the artificial intelligence field and they offer near optimal solutions to large-sized instances for which exact methods cannot be still effective.

1.4. Organization of the Dissertation

The remainder of this dissertation is organized as follows. The second chapter provides a review of the relevant literature dealing with the FLP. This chapter reviews the FLP in general, by considering two classes of problems: the static and the dynamic FLP. In each category, some of the well-known works are discussed namely, exact methods, heuristics and approximate approaches. Literature on the applications of FLP to healthcare is then discussed in the third section of chapter 2 with an emphasis on the used methods and the advantages and disadvantages of each approach.

In this work, we studied the Operating Theater Layout Problem (OTLP) under static and dynamic environment. The third chapter introduces the problem statement for the static OTLP and proposes three Mixed Integer Programming (MIP) formulations to provide exact solutions for small sized instances. Thus, different formulations are presented for the OT layouts, namely the multi-sections, the multi-floors and the multi-rows formulations as well as the computational experiences related to each formulation. While Chapter 4 introduces the dynamic OT Layout with the above three-mentioned formulations and present the proposed MIP to solve the continuous-representation-based on FLP.

In chapter 5, we extend the research on static and dynamic OT Layouts to deal with real life sizes. In this chapter, we develop a Multi Agent System based on the MIP presented in the two previous chapters to solve multi-department layouts. We first present the MIP model for each layout environment (i.e. static and dynamic) and then illustrate the developed multi agent approach based on Master/Slave architecture. Finally, numerical experiments are provided to illustrate the effectiveness of this approach to solve large sized instances.

Chapter 6 investigates approximate approaches and presents a Particle Swarm Optimization method to solve the static multi-department variants while each department is solved as a multi-row layout. We first present the used MIP model, followed by a new developed heuristic to generate initial solutions to encode particles. Finally, we illustrate in details the PSO algorithm and provide a comparison of obtained numerical results with other methods to show the effectiveness of our proposed meta-heuristics.

In Chapter 7, we summarize our research contributions in this dissertation and provide our perspectives on future research direction in the field of healthcare layout.

Chapitre I.

Introduction générale

La conception d'un Bloc Opératoire (Operating Theater (OT)) dans les hôpitaux a été prouvée être un processus complexe. Premièrement, les patients qui ont besoin d'être admis dans les hôpitaux doivent être convaincus de la qualité et la sécurité du fonctionnement de ce lieu mystérieux et symbolique. Deuxièmement, le corps professionnel travaillant aussi dans ces endroits nécessite un environnement architectural et technique optimale pour assurer l'efficacité et de l'efficience de leurs interventions. Enfin, les managers et les gestionnaires sont sous pression croissante pour répondre aux nouvelles normes de qualité imposées par les organismes de réglementation et d'accréditation. Tous ces facteurs rendent la conception du bloc opératoire un projet complexe et d'une importance primordiale pour assurer la satisfaction de toutes les parties prenantes afin de faciliter la circulation des patients et des fournisseurs de soins de santé.

Le travail décrit dans cette thèse vise à fournir des méthodes de modélisation et de résolution pour aider les gestionnaires des hôpitaux avec des outils analytiques pour améliorer l'efficience et l'efficacité de leur conception de bloc opératoire. Dans la section 1.1, nous présentons nos motivations pour mener ce projet de recherche. Dans la section 1.2, nous présentons une introduction du problème et un survol des approches les plus connues pour résoudre le problème d'agencement (FLP) et les fonctions objectifs fréquement utilisées. Dans la section 1.3, nos objectifs de recherche sont présentés. Enfin, dans la section 1.4, nous présentons les grandes lignes de cette thèse.

1.1. Les motivations

Actuellement, le contexte politique, économique et social en France, caractérisé par la recherche à rationaliser les coûts, le contrôle des dépenses de soins, la consommation croissante des services de soins et le vieillissement de la population, encouragent les hôpitaux à repenser leur conception et restructurer leurs organisations, afin de fournir des services de soins de haute qualité à moindre coût tout en maintenant le confort et le bien-être du personnel.

L'importance de la conception de l'hôpital a été prouvée ayant un grand impact sur la sécurité de l'hôpital. Par exemple, un rapport de la Commission d'accréditation des organismes de santé a cité l'environnement physique comme une cause racine de 50% des chutes des patients. Dans une enquête sur les conditions de travail menée par le ministère du travail en

France, il a été indiqué que plus de 67% des employés ne sont pas en mesure de faire leur travail correctement en raison de locaux inappropriés ou d'équipements mal positionnés (Dares [2013]).

En outre, certaines statistiques ont constaté que 28,9% du temps des infirmiers est consacré à la marche. Toutefois, un environnement physique mal conçu crée des conditions latentes, comme le stress du personnel, des accidents et le l'attente des patients, ce qui peut potentiellement entraîner des événements indésirables dans les hôpitaux et d'affecter la qualité des services de soins.

Le bloc opératoire tient une place prédominante dans la performance d'un hôpital. En effet, la capacité de l'hôpital et de son image attractive sont basées dans une large mesure sur la réputation de son bloc opératoire. Les dépenses du bloc opératoire sont plus élevés de la plupart des services dans un hôpital tant en termes de budget d'investissement qu'en termes de fonctionnalité. Le bloc opératoire a également un impact significatif sur la performance d'autres services hospitaliers. En effet, il joue un peu un rôle de pivot dans l'hôpital.

De nombreux travaux de recherche liés aux blocs opératoires sont présents dans la littérature. Ils peuvent être regroupés dans les six domaines principaux et sont illustrés dans la Figure 1.1. Bien que tous les domaines sont interdépendants, notre travail aborde le troisième domaine, où il traite de la conception et de la disposition, qui vise à déterminer l' 'agencement optimal' d'un ensemble de salles au sein de l'espace disponible soumis à des contraintes imposées par le plan du site, la zone de constructions, les exigences des services de soins, et les maitres d'ouvrages. L'importance de ce sujet provient de l'impact de la conception du bloc opératoire sur ses fonctions d'exploitation, la sécurité et la satisfaction des patients et du bien-être des soignants.

Malheureusement, depuis plusieurs décennies, la conception des blocs opératoires a été réalisée par les architectes en utilisant les outils traditionnels basés sur leurs expériences et certaines normes prédéfinies sans utiliser des systèmes intelligents de prise de décisions. En conséquence, la conception finale du bloc opératoire ne peut être optimisée pour faire face à tous les flux de patients, le matériel, le personnel médical et non médical.

À la lumière de ces lacunes, apporter un outil novateur pour la conception du bloc opératoire tenant compte des objectifs de l'hôpital, des aspects d'optimisation, et des normes internationales en matière de réglementation et de l'architecture est devenu un défi ambitieux abordé dans ce travail.

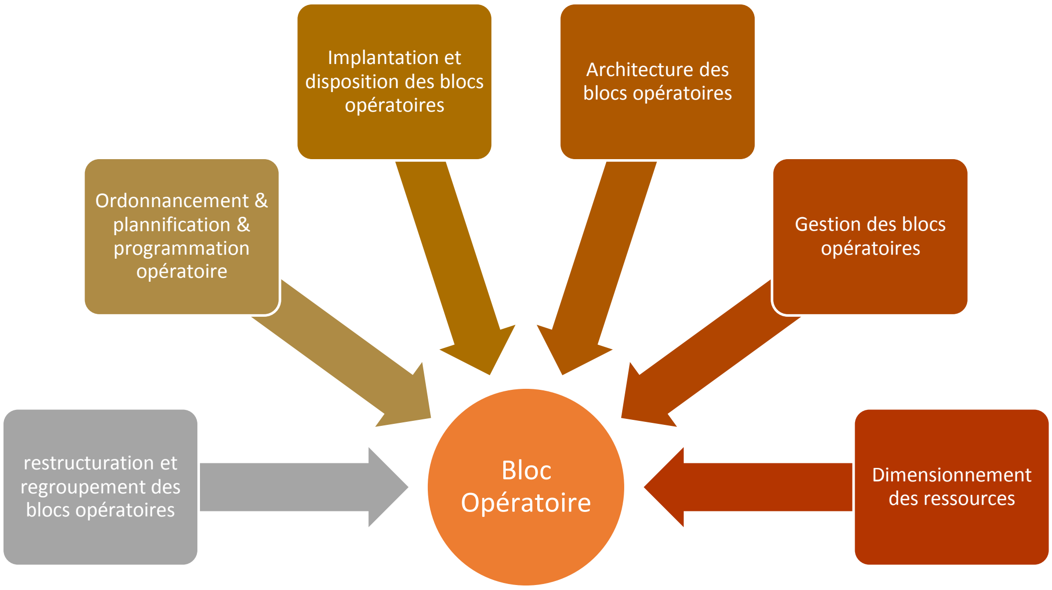


Figure 1.1. Domaines de recherche liés aux blocs opératoires

Nous visons à développer un système de prise de décision qui permet aux décideurs de concevoir un bloc opératoire efficace en introduisant des modifications sur le FLP. Ainsi, la conception du bloc opératoire a été générée en utilisant des modèles MIP en fonction de la représentation de l'environnement (statistique FLP (SFLP) et dynamique FLP (DFLP)) et différentes configurations de conception (structure multi-section, structure multi-étage et structure multi-rangée) pour concevoir un bloc opératoire simple pour les hôpitaux de petite taille.

Pour les plus grands hôpitaux où la structure du bloc opératoire est plus complexe, nous avons introduit l'utilisation des MAS et la PSO pour les différentes représentations de l'environnement et de configurations de conception.

1.2. Les problèmes d’agencement

1.2.1. Introduction

Le problème d’agencement des installations ou le FLP peut être défini comme la détermination de l'organisation physique d'un système de production et de l’arrangement de tout ce qui est nécessaire pour la production de biens ou la prestation de services. Le FLP a de larges applications dans l’industrie, entre autres dans la conception d'hôpitaux, de lignes d'assemblage, les aéroports, entrepôts, bureaux. Il est connu pour avoir un grand impact sur la productivité, les coûts de fabrication, les délais et l'efficacité de l'exécution de son organisation adoptée. Un bon placement des installations contribue à l'efficacité globale des opérations et il peut réduire les dépenses d'exploitation totales de près de 50% (Tompkins et al., [2010]).

De nombreux travaux de recherche ont été publiés dans ce domaine. Le FLP a été défini comme suit :

« *Le FLP est un arrangement de tout le nécessaire pour la production de biens ou de prestation de services. Une installation est une entité qui facilite l'exécution de tout travail. Il peut être une machine, un outil, un centre de travail, une cellule de fabrication, un atelier d'usinage, un département, un entrepôt, etc.* » Heragu (1997)

« *Le FLP est une famille de problèmes de conception impliquant la partition d'une région plane en départements ou domaines de travail de surface connue, de manière à minimiser le coût associé à des interactions projetés entre ces départements.*» Shouman et al. (2001)

« *Le FLP consiste à disposer n installations de différentes tailles à l'intérieur d'un espace donné, qui peut être limitée à la longueur ou la largeur de la zone du site de manière à minimiser le coût du déplacement du matériel et le coût de non-exploitation des zones.* » Lee et Lee (2002)

« *Le FLP est un problème d'optimisation qui cherche à concevoir des structure plus efficace en tenant compte des différentes interactions entre les systèmes de manipulation de matériel et les installations lors de la conception des modèles.*» Shayan et Chittilappilly (2004)

« *Le FLP consiste à déterminer l'organisation physique d'un système de production et de trouver la structure la plus efficace de n installations indivisibles dans n emplacements.* » Singh et Sharma (2006)

A partir de ces définitions et d'autres présentes dans la littérature, le FLP s'intéresse à trouver l'agencement optimal ou le meilleur de n installations dans l'espace disponible sous réserve d'un ensemble de contraintes tout en optimisant un ensemble donné d'objectifs.

1.2.2. Classifications de structure

FLP peut être classé en fonction de différents aspects tels que décrits dans (Drira et al. [2007]) et résumé dans La Figure 1.2, à savoir, l'évolution de l'environnement, la représentation de la structure, les systèmes de fabrication, la configuration de la structure, les formes d'installations, les objectifs du FLP, les contraintes du FLP, etc.

i) L'évolution de l'environnement

Le FLP peut être classé en deux catégories principales en fonction des flux de déplacement, (1) FLP statique (SFLP) et (2) FLP dynamique (DFLP). Le SFLP suppose que les flux entre les installations et les demandes de produits sont constants dans une seule période de temps. Au contraire, le FDLP traite les structures dont l'évaluation et la modification de l'aménagement se produit sur plusieurs périodes. Elle étend le SFLP en supposant que les flux de manipulation et la demande de produits peuvent changer au fil du temps. Dans le DFLP il est nécessaire d'évaluer périodiquement les changements dans la demande de produits pour déterminer la nécessité de réorganiser la disposition, afin de maintenir un bon aménagement de l'installation. (Drira et al., [2007]).

ii) La représentation de l'espace

Le FLP peut également être classé en fonction de la méthode de représentation. Généralement, il y a deux représentations selon Tompkins et al, (2010) :

- *La représentation discrète* : l'espace est représenté par une structure de grille tracée, (comme le montre la Figure 1.3.a) où chaque département est arrondi à un nombre entier de cases ;
- *La représentation continue* : les dimensions du département ne sont pas limitées à une structure de grille, mais plutôt, comme dans la Figure 1.3b, elles peuvent prendre des valeurs non entières.

Nous pouvons dire que la représentation continue est plus précise et plus représentative que celle discrète. La représentation continue augmente la complexité de la résolution du FLP; cependant, il est capable de trouver la 'vraie' solution optimale du FLP.

iii) Le niveau d'abstraction

Les types de FLP sont basés aussi sur les niveaux d'abstraction. Selon Tompkins et al (2010), il y a deux niveaux :

- *La disposition en bloc* : s'intéresse aux flux macro dans l'établissement (la conception montre l'emplacement et la taille de chaque département comme dans la Figure 1.4a)
- *La disposition détaillée* : concerne les flux micro (la conception montre l'emplacement exact de chaque installations, les structures des cloisons, les points d'entrée / sortie (I / O), et la disposition au sein de chaque département, voir la Figure 1.4b).

iv) Les systèmes de production

Les types de FLP varient aussi selon le système de production. Selon Tompkins et al (2010), il existe quatre types comme suit :

- *Agencement de la ligne de production* : ce type est basé sur la séquence de traitement des pièces en cours de production sur une ligne. Le flux de produits se déplace directement à partir d'une station de travail à la station de travail suivante dans la même ligne et il existe souvent plusieurs lignes. (Figure 1.4.a)

- *Agencement de produit fixe* : dans ce type particulier de disposition, le produit ne bouge pas, elle implique le déplacement de postes de travail autour du produit pour effectuer les opérations dessus. (Figure 1.5.b)
- *Agencement en famille de production* : ce type est basé sur le regroupement des postes de travail en entités pour former des familles de produits. Les postes de travail peuvent être regroupés en familles selon des séquences de traitement communes, aux prescriptions de l'outillage, des exigences de manipulation / stockage / de contrôle, etc. (Figure 1.5.c)
- *Agencement de processus* : dans ce type, la disposition d'un département de processus est obtenue en regroupant les mêmes processus ensemble et placer les différents regroupements de processus les uns par rapport aux autres en fonction du flux interdépartemental. (Figure 1.5.d)

v) Objectifs

Le FLP vise à trouver un arrangement efficace sans chevauchement des n installations dans un espace donné. Dans la littérature, la minimisation du coût totale de déplacement et la maximisation du taux de proximité entre chaque deux départements sont souvent les deux objectifs communs utilisés (Meller & Gau [1996]).

- *Minimiser* le coût total du déplacement : nous définissons la fonction basée sur la distance comme suit :

$$\mathbf{Min} \sum_{i} \sum_{i} (f_{ij} c_{ij})\, d_{ij}$$

Où f_{ij} est la fréquence du flux de matériels de l'installation i à l'installation j, c_{ij} est le coût unitaire de déplacement, et d_{ij} est la distance de l'installation i à l'installation j. Cette distance entre le centre géométrique de deux installations peut être calculée avec la formule rectiligne :

- Pour une disposition sur un seul étage :

$$d_{ij} = |x_i - x_j| + |y_i - y_j|$$

- Pour une disposition sur plusieurs étages :

$$d_{ij} = |x_i - x_e| + |y_i - y_e| + \varepsilon|F_i - F_j| + |x_e - x_j| + |y_e - y_j|$$

Où (x_i, y_i); (x_j, y_j); (x_e, y_e) sont respectivement les coordonnés de l'installation i, de l'installation j et de l'ascenseur e partant du centre du repère (0,0).

F_i = étage (i) où l'installation i est localisée.

F_j = étage (j) où l'installation j est localisée.

$\boldsymbol{\varepsilon}$ = facteur pour le déplacement vertical et le temps d'attente.

La distance peut être exprimée considérant les points d'entrée et de sortie (pick-up and drop-off). Dans ce cas, la distance parcourue est calculée à partir de la sortie de l'installation i à l'entrée de l'installation j :

$$\boldsymbol{d_{ij} = |x_i^O - x_j^I| + |y_i^O - y_j^I|}$$

Où $(\boldsymbol{x_i^I}, \boldsymbol{y_i^I})$; $(\boldsymbol{x_i^O}, \boldsymbol{y_i^O})$ représentent les coordonnés du point d'entrée et le point de sortie respectivement.

- *Maximiser le taux de proximité* : cette fonction peut être exprimée comme suit :

$$\mathbf{Max} \sum_i \sum_i (\boldsymbol{r_{ij}})\, \boldsymbol{x_{ij}}$$

Où $\boldsymbol{r_{ij}}$ représente la valeur numérique du facteur de proximité entre les installations i et j, et $\boldsymbol{x_{ij}}$ est égal à 1 si les installations i et j sont adjacents, et 0 sinon basées sur la distance qui les sépare.

- *La somme pondérée de plusieurs fonctions objectives* :

Ayant Connaissance des avantages et des inconvénients des deux objectifs (Meller & Gau [1996]), certains auteurs ont combiné les deux objectifs sous une forme pondérée comme suit :

$$\mathbf{Min}\, \boldsymbol{\alpha} \sum_i \sum_i (\boldsymbol{f_{ij} c_{ij}})\, \boldsymbol{d_{ij}} - (\mathbf{1} - \boldsymbol{\alpha}) \sum_i \sum_i (\boldsymbol{r_{ij}})\, \boldsymbol{x_{ij}}$$

Où α est le poids des fonctions objectives, Meller & Gau (1996) ont étudié comment régler α pour optimiser la fonction de somme pondérés.

vi) Les méthodes de résolution

Tout type de FLP, que ce soit avec formulation SFLP / DFLP ou représentation Discrète / continue est résolu en utilisant trois types d'approches de résolution :

- *Les méthodes exactes :*

En théorie, les algorithmes exacts sont utilisés pour obtenir une solution optimale globale du FLP pour garantir l'optimalité de la disposition finale. Malheureusement, ils sont extrêmement chronophages et ne peuvent traiter que les problèmes avec de très petites tailles (Osman et al. [2002]) qui sont loin de la réalité des problèmes industriels (30-40 départements). Les méthodes bien connues pour ces types sont ceux utilisées pour résoudre le problème d'affectation quadratique, les méthodes basées sur la théorie des graphes et les modèles de MIP.

- Problème d'affectation quadratique (QAP)

Le problème d'affectation quadratique est un cas particulier de la FLP. Il assume des emplacements fixes et connus, des surfaces égales pour chaque installation et une affectation une-à-une entre les installations et les emplacements. Une formulation typique de QAP est donnée comme suit selon Koopmans et Beckmann (1957) :

$$\min \sum_{i}^{n} \sum_{j}^{n} \sum_{k}^{n} \sum_{l}^{n} c_{ijkl} x_{ik} x_{jl}$$

Où :

$$\sum_{i}^{n} x_{ik} = 1 \qquad k = 1, \ldots, n$$

$$\sum_{j}^{n} x_{jl} = 1 \qquad l = 1, \ldots, n$$

$$x_{ik} \begin{cases} 1 & \text{si l'installation i estaffectée à l'emplacement k} \\ 0 & \text{sinon} \end{cases} \quad i, k = 1, \ldots, n$$

Où c_{ijkl} est le coût supporté pour affecter l'installation i à l'emplacement k et l'installation j à l'emplacement l.

Les algorithmes exacts pour résoudre le QAP comprennent des approches fondées sur le Branch & Bound (séparation et évaluation), les plans sécants et la programmation dynamique. Les algorithmes Branch & Bound sont ceux qui ont eu le plus de succès, mais ils sont encore incapables de résoudre des instances de grandes tailles.

- La théorie des graphes

Dans les approches de théorie des graphes, il est supposé que les facteurs de proximité entre les installations sont connus. Chaque installation est alors représentée par un nœud dans un graphe et les facteurs d'adjacence sont représentés par un arc reliant deux installations voisines. La fonction objectif dans ce cas est de maximiser le taux de proximité entre les installations.

Pour développer une disposition en utilisant des approches de théorie des graphes, nous créons d'abord un graphe à partir de la matrice donnée des relations entre installations que nous transformons après à un graphe dual et enfin nous générons une disposition en bloc à partir de ce graphe dual. Comme dans le QAP, l'optimalité ne peut pas être garantie pour les problèmes de grandes tailles. Comme résultat, beaucoup d'heuristiques de construction basées sur le modèle de théorie des graphes ont été développées. Hassan & Hogg (1987) ont présenté une analyse approfondie de ces heuristiques.

- Programmation à nombre entiers mixtes

Comparé à d'autres méthodes, la programmation en nombres entiers mixtes (MIP) est une nouvelle approche à l'origine présentée par Montreuil (1990) pour résoudre le FLP. Cette approche utilise un objectif basé sur la distance et suppose une représentation continue du FLP. Le MIP peut résoudre les problèmes avec des départements à surfaces égales et inégales en spécifiant la localisation et l'orientation des installations avec l'utilisation de variables binaires pour empêcher le chevauchement de ces installations.

- *Les heuristiques* :

Prenant en considération le déficit de QAP, les modèles de la théorie des graphes, ou le MIP de garantir des solutions optimales pour résoudre les problèmes de grandes tailles, les chercheurs ont fait des efforts pour trouver des solutions approximatives en mettant en œuvre des approches heuristiques. Ces heuristiques peuvent être classées en deux catégories : les heuristiques de construction et les heuristiques d'amélioration. Les heuristiques de construction commencent successivement par la sélection et la localisation d'une installation jusqu'à ce que la disposition soit terminée pour obtenir une solution unique, tandis que les heuristiques d'amélioration sont basées sur l'amélioration d'une disposition initiale en utilisant l'algorithme d'amélioration jusqu'à ce qu'il n'y ait plus d'améliorations à proposer en utilisant des échanges par paires et d'autre type d'échanges. Ces catégories utilisent soit des approches basées sur la proximité ou à base de distance.

- *Les Métaheuristiques* :

Osman (1995) a défini les métaheuristiques comme « un processus de génération itérative qui guide une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter l'espace de recherche, les stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver de manière efficace des solutions quasi-optimale». Ils sont la classe la plus récente des méthodes approchées qui guident le processus de recherche des méthodes classiques de construction et d'amélioration pour éviter l'optimum local. Elles sont largement utilisées pour résoudre des problèmes complexes dans différents domaines industriels, Osman et Kelly (1996). Les Métaheuristiques comprennent entre autres la recherche tabou (TS), algorithme génétique (GA), l'optimisation de colonie de fourmis (ACO), recuit simulé (SA), optimisation par essaim de particules (PSO), les Hyper-heuristiques. Une étude récente des travaux sur métaheuristiques appliquées au FLP est fournie dans Barsegar (2011) tandis que l'origine des métaheuristiques est fournie dans Osman (1995).

1.3. Les objectifs de recherche

Bien qu'il existe un nombre important de travaux de recherche pour résoudre le FLP en considérant différentes disposition dans différents domaines d'application, les recherches sur la structure de soins et de santé en général et en particulier le bloc opératoire sont encore limitées malgré l'importance de tel environnement. Ce travail vise à combler cette lacune et de proposer des approches différentes pour résoudre le FLP appliqué au bloc opératoire.

Les objectifs de notre recherche peuvent être décrits comme suit:

Notre objectif de recherche principal est de développer un système intelligent d'aide à la décision pour les hôpitaux afin d'offrir un outil aux gestionnaires qui est à la fois facile, compréhensible, fiable et financièrement viable pour concevoir leurs blocs opératoires en tenant en compte :

- les besoins et les contraintes des équipes de chirurgiens et d'anesthésistes
- la disponibilité des infirmières
- la relation entre les différentes salles et services du bloc opératoire
- les procédures d'approvisionnement des divers équipements
- l'activité des brancardiers
- les ressources des salles de repos.

La fiabilité de l'outil développé est assurée en suivant les normes internationales régissant la conception des établissements de soins et de santé. Certaines des références détaillant ces normes sont présentés ici :

- AIA (American Institution of Architecture), Guidelines for Design and Construction of Health Care Facilities;
- FGI (Facility guideline institution), Guidelines for Design and Construction of Health Care Facilities;
- JCAHO (Joint Commission on Accreditation of Healthcare Organizations) Planning, Design, Construction of healthcare facilities;
- ICRA (Infection Control Risk Assessment), Matrix of Precautions for Construction & Renovation;
- ASHRAE (American Society of Heating, Refrigerating, and Air-Conditioning Engineering), ventilation standard for healthcare facility; etc.

Tandis que l'aspect de la rentabilité est obtenue en utilisant des algorithmes d'optimisation, qui sont exécutés en arrière-plan pour fournir l'optimalité pour les instances de petite taille ou une approximation de l'optimale pour la conception des blocs opératoires dans le cas de problèmes de plus grande taille sous différentes spécifications telles que :

- Le mode d'hospitalisation planifiée (classique ou ambulatoire)
- Le nombre d'interventions, leurs types et leurs durées moyennes
- Le nombre de salles d'opération et de pré-anesthésie
- Le nombre de personnes travaillant dans un bloc : chirurgiens, des anesthésistes, médecins et infirmières ; etc.

Le deuxième objectif fixé de ce travail de recherche est de fournir des méthodes exactes pour résoudre à l'optimalité trois formulations différentes sous une demande de soins statique et dynamique. Les trois formulations considèrent le FLP basé sur la représentation continue et représentent la façon avec laquelle les décideurs aperçoivent le bloc opératoire final, soit comme une structure en plusieurs sections sur un étage, comme une structure en plusieurs étages avec l'utilisation des ascenseurs ou comme une structure en plusieurs rangées en utilisant des couloirs pour relier les différentes parties du bloc opératoire.

Le troisième objectif est de résoudre des problèmes avec des tailles de la vie réelle pour les blocs opératoires ayant plus de trente salles tout en assurant l'efficacité de la solution proposée. Cet objectif est atteint en utilisant deux approches approximatives à savoir un système multi-agents et un algorithme d'optimisation par l'essaim de particules. Ces deux méthodes sont des dérivées de l'intelligence artificielle et ils offrent des solutions quasi-optimales pour des instances de grandes tailles pour lesquelles les méthodes exactes ne sont toujours pas efficaces.

1.4. Organisation de la thèse

Le reste de cette thèse est organisé comme suit : Le deuxième chapitre présente un état de l'art des travaux pertinents traitant le FLP. Ce chapitre examine le FLP en général, en tenant compte de deux classes de problèmes : la FLP statique et dynamique. Dans chaque catégorie, certains des travaux connus sont analysées à savoir, les méthodes exactes, les heuristiques et les approches approximatives. La littérature sur l'application du FLP aux systèmes de soins et de santé est ensuite discutée dans la troisième section du chapitre 2 en mettant l'accent sur les méthodes utilisées et les avantages et les inconvénients de chaque approche.

Dans ce travail, nous avons étudié l'OTLP dans un environnement statique et dynamique. Le troisième chapitre présente la problématique pour l'OTLP statique et propose trois formulations MIP pour fournir des solutions exactes aux instances de petites tailles. Ainsi, différentes formulations sont présentées pour la conception du bloc opératoire, à savoir, structure multi-section, multi-étages et multi-rangées, de même que les expérimentations et les résultats numériques liés à chaque formulation. Alors que le chapitre 4 présente la conception du bloc opératoire sous environnement dynamique avec les trois formulations mentionnés ci-dessus et présente les modèles MIP proposés pour résoudre le FLP sous la représentation continue.

Dans le chapitre 5, nous étendons la recherche sur les OTLP statiques et dynamiques pour faire face aux problèmes de tailles de la vie réelle. Dans ce chapitre, nous développons un système multi-agents basé sur le MIP présenté dans les deux chapitres précédents pour résoudre un problème à plusieurs départements (structure multi-départements). Nous présentons d'abord le modèle MIP pour chaque environnement (statique et dynamique), puis nous illustrons l'approche multi-agents développée basée sur une architecture maître / esclave. Enfin, des expériences numériques sont fournies pour illustrer l'efficacité de cette approche pour résoudre les instances de grandes tailles.

Le chapitre 6 étudie les approches approximatives, et présente une méthode d’optimisation d’essaim de particules pour résoudre la variante multi-département sous environnement statique du FLP, tandis que chaque département est résolu comme une structure multi-rangée. Nous présentons d'abord le modèle MIP utilisé, suivie d'une nouvelle heuristique développée pour générer des solutions initiales afin d’encoder les particules. Enfin, nous illustrons en détail l'algorithme PSO et nous fournissons une comparaison des résultats numériques obtenus avec d'autres méthodes pour démontrer l'efficacité de notre métaheuristique proposée.

Dans le chapitre 7, nous résumons nos contributions dans cette thèse et présentons les perspectives de recherche dans le domaine de l’agencement des établissements de soins.

Chapter II.

Literature review

2.1. Introduction

The Facility Layout Problem (FLP) is a common industrial problem of allocating facilities to specific designated locations. The FLP seeks to determine the 'most efficient' placement of facilities (facilities) within a designated section of a plant, subject to some constraints imposed by the physical departmental areas, system operating requirements and the desire of the decision-makers [Meller et al. (1996)]. It deals with the physical arrangement of operating units into specific locations/spaces to reduce the total cost of material handling (flow of products) between spaces and to provide the best support for efficient production. Therefore, the allocation affects the productivity and efficiency of the industrial organization. In fact, a good placement of facilities contributes in general to the overall efficiency of operations and can reduce the total operating expenses up to 50% [Tompkins et al. (2010)].

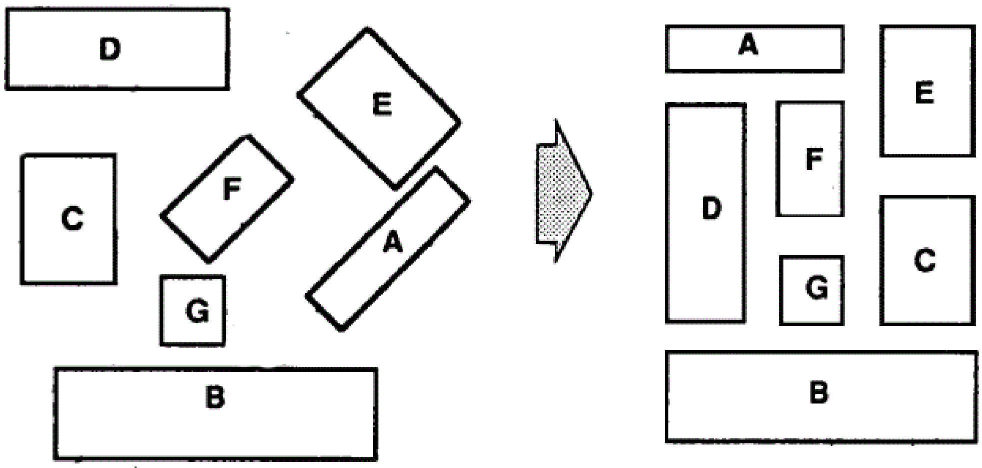


Figure 2.1. An illustration of a Facilities Layout Problem

Figure 2.1 provides an illustration of typical set of operating units of different rectangular sizes (denoted by A,.., F) to be placed on rectangular plane according to certain desired criteria and restrictions.

The FLP has practical and theoretical importance. It has been widely used by researchers to design floor layouts of offices in: buildings, airports, warehouses and hospitals among others, [Osman et al. (2002)]. The FLP has attracted lots of research attention which are demonstrated through the various classifications of methods that differ according the type of applications domain: manufacturing systems; service configurations, objective functions, shape of facilities and other operating constraints, [Drira et al. (2007)]. One of the most broadly used classifications of methods is based on the material handling of flows and layouts, namely: (1) Static Facility Layout Problem (SFLP) and (2) Dynamic Facility Layout Problem (DFLP).

In the remainder of this chapter, we shall provide a detailed literature review of the FLP research literature based on the above classification. In Section 2.1, we shall review the exact and approximate methods for the Static Facility Layout Problem and while in Section 2.2 we shall review that of the dynamic Facility layout problem. Finally, please note that the facilities and departments will be used interchangeably in the text.

2.2. Static Facility Layout Problem

In the SFLP, it is assumed that there is a deterministic flow of products/people between facilities/activities and a fixed number of locations to be determined over a single-period planning horizon. SFLP has been solved using different solution approaches classified either exact or approximate methods.

2.2.1. Exact Methods

Exact algorithms for the FLP are those methods developed to obtain, in theory, optimal solutions (or known as the best feasible solutions). SFLP has been modelled using well-known optimization models, namely: i) Quadratic Assignment Problem (QAP) [Koopmans et al. (1957)], ii) Mixed Integer Programming [Montreuil (1990)] and iii) Graph Theoretic Approaches [Seppänen & Moore (1970)].

i) Quadratic Assignment Problem.

The QAP was first developed by Koopmans Beckman (1957), in which facilities are assumed to have equal areas, and locations are fixed and known a priori. A typical QAP formulation is as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} c_{ijkl} x_{ik} x_{jl} \tag{2.1.1}$$

Subject to:

$$\sum_{i=1}^{n}\sum_{k=1}^{n} x_{ik} = 1, k = 1, \dots, n \tag{2.1.2}$$

$$\sum_{j=1}^{n}\sum_{l=1}^{n} x_{jl} = 1, l = 1, \dots, n \tag{2.1.3}$$

$$x_{ik};\ x_{jl}\ \geq 0\ and\ Integer \tag{2.1.4}$$

Where c_{ijkl} is the material handling cost per unit of flow between facilities i and j if they were located at k and l respectively, and $x_{ik} = 1$ if facility i is located at location k, and 0 otherwise.

The QAP is theoretically challenging problem, it belongs to the class of NP-Hard problem [Sahni and Gonzalez (1976)]. This challenge makes it one of the most difficult combinatorial optimization problems. The simplicity of QAP model draws the attention of researchers to use it for modeling a variety of applications. However, the QAP model suffers limitations in finding optimal solutions to large-sized instances. Despite such challenge, exact solution methods can be grouped into three main categories: the Branch and bound procedures [Gilmore (1962), Lawler (1963)], the cutting plane techniques [Bazaraa and Sherali (1982), Burkard and Bonninger (1983)], and the Dynamic programming methods [Christofides and Benavent (1989)].

a) Branch and Bound

Branch and bound (B&B) is one of the most important and used method to achieve optimality for QAP. B&B algorithm receives its name from the way it is executed. First, a simple lower bound can be generated by relaxing the integrality of constraint (2.4) i.e., allowing the variables to take values between zero and 1 ($0 \leq x_{ij} \leq 1$). This linear relaxation generates an infeasible solution with an objective function value called lower bound. The branch and bound process proceed by creating two sub-problems by choosing one of the fractional variable and fixing it either to zero or to 1 by adding two constraints (x_{ij}= 0) and ($x_{ij} = 1$), hence creating two branches (Sub-problems are created). The search tree is continues by repeating the decomposition until the first feasible solution is obtained. The first feasible solution is declared an upper bound for the problem. The search backtracks to the next infeasible solution by considering another real variable to branch on. If any of the new branches (children) has a lower bound of value smaller the current upper bound, it is fathomed or bounded, otherwise the search continue to find a better upper bound. The B&B procedure requires a criterion to select a variable to branch on, (highest real value) to assign a facility to a location. The B&B procedure terminates when the search process backtracks to the first node of the tree search and the best feasible solution (upper bound) is declared as the obtained optimal solution. It is often less difficult to obtain a feasible solution than confirming the optimality of a solution. The latter requires a complete exploration of the search space but the former requires executing the tree search until a feasible solution is obtained. The gap between the upper and the lower bounds determines the quality of the obtained feasible solution.

Gilmore (1962) introduced for the first time the Branch and Bound procedure and solved a QAP of size n=8. In 1988, Hahn et al. (1998) presented a B&B procedure based on the Hungarian method to solve QAP. In each stage, they used a dual procedure for the QAP structures to fully obtain an equivalency of the original QAP to calculate lower bounds on the optimal solutions. This approach allowed them to solve to optimality problems with n = 22 facilities.

Brixius & Anstreicher (2001) described a B&B algorithm using a convex quadratic programming relaxation for problem and solved instances with n = 24. The used approach is based on the Frank-Wolfe (FW) algorithm using dual information to estimate the effect of fixing an assignment to create a sub-problem in each node in the B&B tree. In 2002, Anstreicher et al. (2002) presented a B&B using a federation of geographically distributed resources known as a computational grid to improve the algorithm solving ability to n = 30 facilities.

Solimanpur & Jafari (2008) presented a new nonlinear mixed-integer programming model for a two-dimensional facility layout problem aiming to minimize the total distance travelled by the material in the shop floor. Then, the technique of Kusiak (1990) was used to linearize the MIP model, and finally, a branch-and-bound approach was developed to optimally solve small-sized and medium-sized instances of sizes.

b) Cutting Plane

The cutting plane technique is an alternative to branch and bound to solve integer programs, which consists to add constraints to a linear program until the optimal basic feasible solution takes on integer values. The Cutting Plane methods can be divided into three classes: traditional cutting plane methods; polyhedral cutting-plane; and branch-and-cut methods. Bazaraa and Sherali (1980) were the first to introduce the cutting plane method, but no satisfactory results were achieved because of its slow convergence. The method insured optimality for just very small-sized problems (n = 6), thus it was appropriate only for small instances [Kaufman and Broeckx (1978), Bazaraa and Sherali (1982) and Burkard and Bonniger (1983)]. However, the cutting plane method still has been used with some heuristics; Mixed-Integer Linear Programming (MILP) and benders decomposition. For instance, Maniezzo & Colorni (1999) solved QAP instances with n = 25 facilities by a heuristic using the cutting plane algorithm.

c) Dynamic Programming

The dynamic-programming algorithm consists of solving a problem of size 'n' by solving its smaller sub-problems of size of 1, 2 …n-1. The method progresses from solving small sub-problems of size k, moves to larger ones of size k+1 and dynamically add one by one until a finally solution to the original problem is obtained. There have been fewer applications for dynamic programming (DP) methods than the cutting plane and the Branch and Bound method in the literature. Christofides and Benavent (1989) were the first to use a DP approach to solve a special case of QAP (the tree QAP). They used a Lagrangian relaxation of an integer programming formulation to compute a lower bound for the problem. They solved the relaxed problem using a DP algorithm. The method gave solutions for the QAP instance in the library (QALIB) with size up to n = 25. Marzetta & Brüngger (1999) also solved the QAP with a DP method using parallel processing algorithm to provide optimal solution the unsolved instance NUG25 in the QALIB at that time.

Recently, Nyberg & Westerlund (2012) used an exact discrete linear formulation of the problem with a reduced number of bilinear terms to n^2. They managed to solve some of the previously unsolved instances of sizes n = 32 and n = 64. The same authors Nyberg et al. (2013) improved the discrete formulation by reformulating the bilinear terms in different manners to reduce the processing time and to solve the last unsolved instances from the QALIB by exploiting the symmetry of the QAP matrices. Fischetti et al. (2012) solved the large instance in the QALIB (n = 128). Their method was based on three steps: exploiting the symmetry in the flow and distance matrices; generating a suitable MILP using both an off-the-shelf (black-box) MILP solver and an ad-hoc branch-and-cut scheme; and finally, exploiting a Facility-Flow Splitting Scheme.

ii) Mixed Integer Programming

The Mixed Integer Programming (MIP) is an extension of the discrete QAP to the continuous layout space where the locations of facilities are not predefined with known distance. The distances become continuous variables and depend on the locations of facilities. The MIP formulation differs from the QAP one in few key aspects: the size and the orientation of each facility; a multi-objective function which includes material traveling cost, adjacency between facilities, and construction costs among others. The above difference makes the problem additionally non-linear mixed integer problems due to its unknown distance between facilities.

The MIP was originally presented for the FLP in 1990 by Montreuil (1990) using a distance-based objective while considering facilities with unequal areas. In this model, the orientation and location of facilities are decision variables. A number of inequalities using binary integer variables are formulated to avoid overlapping facilities.

Papageorgiou & Rotstein (1998) developed a MILP formulation to find optimal plant layout in a two-dimensional continuous space in a single-floor. Patsiatzis & Papageorgiou (2002) extended the work of Papageorgiou & Rotstein (1998) and presented a MIP formulation to design a multi-floor layout for simultaneously determining the number of floors, land area, equipment–floor allocation, and detailing the layout of each floor while minimizing the total cost of material transportation and adjacency requirement between departments. Konak et al. (2006) presented a MIP formulation for the block layout problem with unequal departmental areas located in flexible bays. The main objective was to minimize the departmental material handling cost. They have narrowed their model by using a *p*-position strategy for symmetry breaking to remove three-fourths of the symmetric solution. This model was extended by Jaafari et al. (2009) to deal with the multi objectives FLP. This formulation aims to both minimize departmental material handling cost and maximize closeness rating. Krishna et al. (2009) further addressed a FLP with two-floor and unequal departmental areas using a mixed integer programming formulation. They developed a multi-objective function to minimize the material handling cost and to maximize the closeness of rating. This model determined the position and the number of elevators with consideration of conflicting objectives simultaneously.

More recently, Hathhorn et al. (2013) developed a multi-objective MIP model for multi-floor FLP to minimize both the material handling cost and the total facility building cost. They proposed a lexicographic ordering technique to handle the multiple objectives. Zhang & Che (2014) presented a new formulation of FLP, in which the one-floor or multi-floor building has already been constructed, and the specific room layout inside has been separated into several blocks. They used a MILP formulation to assign a certain number of rooms to a given number of departments while maximizing the utilization rate of the rooms.

iii) Graph Theoretic Approaches

While the objectives of MIP and QAP were to minimize the material handling cost within a plant, the objective of the Graph-Theoretic (GT) approach is to maximize the weighted sum of desirable measures among adjacent facilities (departments). In graph-theoretic approaches, it is assumed that the closeness ratings (desirable measures) between departments are known. Each department is then represented by a node in a graph and department adjacency relationships are represented by an arc connecting the two adjacent departments. The area and shape of the departments are ignored, and the objective function is to maximize the weight sum of adjacencies between department pairs, Osman (2006). To develop a facility layout using the graph-theoretic approach, the following steps are normally followed: *first,* an adjacency graph from department relationships is developed, *second* an optimal weighted maximal planar sub-graph is obtained, and finally a block layout is generating from the dual of the optimal sub-graph.

Seppänen and Moore (1970) were the first to introduce the graph theoretic method for FLP. They proposed a solution based on a string grammar. Several research works have been published using this graph approach including Carrie et al. (1978), Foulds & Robinson (1978), Hassan & Hogg (1991), Al-Hakim (1992) and Leung (1992). In general, the graph approach is still difficult to address the problem properly. It needs to use various methods; heuristics/exact to construct a maximally weighted adjacency graph and then to produce optimal bock-layout. However, Osman et al. (2002) introduced an Integer Linear Programming (LP) model of the weighted maximal planar graph problem, and proposed a LP-based heuristics to solve instance up to 100 nodes.

Finally, Kim & Kim (1995) only considered the unequal-sized FLP and proposed a new graph theoretic approach with the objective of minimizing total transportation distance. In their method, the authors assumed that facilities shapes are not fixed, and generated firstly an initial layout by constructing a planar adjacency graph and then improved the solution by changing the adjacency graph.

2.2.2. Heuristic methods

The weakness of exact methods in solving real-life large-sized instances led to the development of approximate methods (or Heuristics). Kusiak and Heragu (1987) classified heuristics for FLP as: *construction* algorithms and *improvement* algorithms. A construction algorithm constructs a feasible solution by assigning facilities one by one to a location according to a well-defined greedy criterion until a complete layout is obtained. An improvement algorithm starts with an initial solution, usually randomly generated (or constructed) and then an exchange process among facilities is conducted to generate a new neighboring solution. The new neighbor is evaluated and accepted to replace the current solution if it improves the objective function. The improvement process stops when the current solution cannot further be improved. The current solution is called a local optimum solution.

The **Deltahedron** was one of the first constructive heuristics, introduced by Foulds & Robinson (1976) and Foulds & Robinson (1978). It generates a final layout in a two-phase approach. In the first phase, a weighted maximal planar layout is generated from the adjacency graph. It starts with the three facility-nodes that have the highest sum of weights; it then adds one at a time a facility-node that maximizes the total sum of adjacency weights of the partially constructed planar graph. The addition of node-facility process continues until a final weighted maximal planar graph is obtained. In the second phase, a floor plan layout of facilities is then derived from the generated maximal weight planar graph. Several extensions of Deltahedron approaches were developed. For instance, Goetschalckx (1992) proposed the **SPIRAL** concept to construct an adjacency graph, where tuples' values quantify the relationship between each two departments.

MATCH is also an interactive adjacency-based constructive heuristic presented by Montreuil et al. (1987). It aims to solve b-matching problems by using a discrete representation and integer programming. **CRAFT** (Computerized Relative Allocation of Facilities Technique) was proposed by Armour & Buffa (1963) as an improvement-type heuristic, which proceeds by implementing two-way or three-way exchanges of the centroids of non-fixed departments. For each exchange, an estimated reduction in cost is calculated and the exchange with the largest estimated reduction is selected.

NLT (Nonlinear optimization Layout Technique) is a construction-type distance-based algorithm introduced by Van Camp et al. (1992) based on non-linear programming techniques. In this heuristic, an exterior point quadratic penalty function method is used to transform the constrained model to an unconstrained form. Hassan et al. (1986) developed a heuristic named **SHAPE**, which uses a distance-based objective. The department placement is based on each department's flows and a user-defined critical flow value, and begins at the center of the layout. Subsequent departments are placed based on the objective function value increase when placing the department on each of the layout's four sides.

Chen & Sha (2005) presented a five-step heuristic to solve multi-objective FLP. First, the heuristic generates an initial solution; then it uses a multi-pass halving and doubling procedure to construct a paired comparison matrix. Third, consistency of this matrix is identified; then the inconsistent matrix is transformed into a consistent one; and finally the heuristic applies a geometric mean method to generate the objective weights and obtains the facility layout solution.

Inspired from the Very Large Scale Integration (VLSI) design literature, Liu & Meller (2007) proposed a combination of the MIP-FLP model and the Sequence-Pair Representation, which aims to reduce the number of binary variables in order to be solved for large-sized instances. To combine the MIP-FLP model and the Sequence-Pair Representation, the authors used a Genetic Algorithm-based heuristic to search the sequence-pair solution space in order to set some of the binary variables to ones for usage in the MIP-FLP model.

Bozer & Wang (2012) proposed a heuristic to prevent the large number of binary variables engendered by non-overlapping constraints. The presented heuristic utilizes a graph-pair representation technique to relax integer constraints and uses a simulated annealing algorithm to drive the layout improvement procedure.

2.2.3. Approximate methods

a) Metaheuristics

Metaheuristics are the most recent class of approximate methods that guide the search process of classical construction and improvement methods to escape local optimum. They widely used to solve complex problems in different industrial fields, Osman and Kelly (1996). Meta-heuristics include tabu search (TS), genetic algorithm (GA), ant colony optimization (ACO), simulated annealing (SA), particle swarm optimization (PSO), Hyper-heuristics, among others. A recent survey of works on metaheuristics applied to FLP is provided in Barsegar (2011) while the origin of meta-heuristics is provided in Osman (1995). In the next paragraphs, we shall review successful approximate methods for solving the facility layout problems.

Kothari & Ghosh (2013) addressed the single row FLP using ***tabu search*** approach combined with two different local search methods. The first involves an exhaustive search of the 2-opt neighborhood by interchanging the positions of exactly two facilities and the second involves an exhaustive search of the insertion neighborhood by removing a facility from its position in the permutation and re-inserting it at another position in the permutation. This approach improved previously the best-known solutions for 23 out of the 43 large sized SRFLP benchmark instances.

García-Hernández et al. (2013) presented an interactive ***genetic algorithm*** for the Unequal Area FLP allowing the participation of the decision maker to guide the search process. At each iteration, the algorithm is adjusted according to the evaluation of the decision maker of the representative solution, which is selected using a c-mean clustering method from the total population of solutions.

Kulturel-Konak & Konak (2011) presented an ***ant colony optimization*** approach for the unequal area FLP with flexible bays, using a dynamic penalty approach to maintain the search in the feasible solution spaces. A local search procedure to improve the performance is implemented based on random swap and insert operators. The method is reported to improve the best-founded solutions in the literature by approximately 17%.

Şahin (2011) developed a ***simulated annealing*** to solve the bi-objectives FLP. The first objective is to minimize the total material handling cost while the second objective is to maximize the total closeness rating scores. The author assigned different weights to the two objectives to evaluate their impact on the quality of the solution.

Samarghandi et al. (2010) used a ***particle swarm optimization*** to solve the single row FLP. The authors employed a factoradic coding and decoding techniques to present each feasible solution by a number and to enable the PSO to explore the continuous solution space. Afterwards, the algorithm decodes the solutions to its respective feasible solution in the discrete feasible space and returns the solutions.

b) Intelligent methods

The origin of intelligent methods came from artificial intelligence, and they have been applied to the optimization context to solve complex problems. Intelligent methods are also a class of approximate methods but less used than meta-heuristics for the FLP. It includes the artificial immune system (AIS), fuzzy systems (FS), multi-agent systems (MAS), etc. Ulutas & Kulturel-Konak (2012) addressed the unequal area FLP with flexible bay structure (i.e. a continuous layout representation allowing the facilities to be located only in parallel bays with varying widths) by using an ***artificial immune system***. A clonal selection algorithm has been developed for the flexible bay structure with a new encoding as well as a new technique to handle the dummy departments dummy departments used to fill the empty space in the facility area.

Azadeh et al. (2014) developed an integrated algorithm based on ***fuzzy*** simulation, ***fuzzy*** linear programming, and ***fuzzy*** data envelopment analysis to deal with a special case of workshop FLP with ambiguous environmental and health indicators. The developed algorithm is used for simulating and ranking a set of layout alternatives generated by VIP-PLANOPT (a computer aided layout planning tool).

Tarkesh et al. (2009) used a society of virtual ***intelligent agents*** to solve FLP. They introduced the concept of emotion for the first time in operational research where each agent models a facility, has proprieties such as money and emotions. Properties are adjusted during the agent's interactions, which are defined by a market mechanism where the richer agents can pay more money to obtain better locations and less for the less interesting locations.

Jiang & Nee (2013) proposed an ***Augmented Reality*** based FLP for an interactive 3D real-time facilities planning composed of four modules. First, the user interaction module, where camera tracks the virtual space and detects the facilities edge. Second, the modelling module, where users can move, add or remove facilities while respecting the criterions and constraints defining the specific requirements of the layout. Third, the evaluation module, where the respect of criterion and constraints is verified and finally, the optimization module, where an Analytical Hierarchy Process–Genetic Algorithm method for automatic layout planning is adopted to optimize the final shop-floor layout.

2.2.4. SFLP Summary

The Static facility layout problem deals with manufacturing systems where the flows between departments and product's demands do not change or nearly constant over time. It is usually considered for a single period planning horizon. This approach is not the same in all manufacturing systems, which are characterized by varying input data during the planning horizon, thus leading to inadequacy of SFLP as a modeling tool. Therefore, dynamic facility layout problem has been introduced to cater for this inadequacy.

2.3. Dynamic Facility Layout Problem

The Dynamic Facility Layout Problem (DFLP) is concerned with the design of multi-period facility layout plans. We distinguish two types of planning horizon: fixed or rolling one. In a rolling horizon plan of m periods, a plan is initially made for a cycle of m periods, but at the end of the first period, the whole data is updated, and a new plan is updated another cycle of m periods. In contrast, a fixed planning horizon, a plan is initially made for the cycle of m periods and a new plan is made at the end of first cycle. Rosenblatt (1986) was the first to discuss the solution techniques for the DFLP based on a Dynamic Programing (DP) model. Reviews on dynamic layout problems and solution approaches are given in Kulturel-Konak (2007) & Balakrishnan et al. (1998). In Balakrishnan et al. (1998), the authors summarize the different algorithms used to solve the DFLP for equal and unequal department's size.

The majority of research in DFLP assumes that facilities have equal sizes and deterministic material flows. This type of formulation is an extension of the well-known quadratic assignment problem (QAP) used for the SFLP with time consideration [Balakrishnan et al. (1992)]:

$$\text{Min} \sum_{t=2}^{P} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{m=1}^{N} A_{tijm} Y_{tijm} + \sum_{t=1}^{P} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{m=1}^{N} C_{tijkm} X_{tij} X_{tkm} \tag{2.2.1}$$

Where:

Y_{tijm} : is a binary 0 or 1 variable for moving ***i*** from ***j*** to ***m*** in period ***t,***

A_{tijm} : is a fixed cost of moving ***i*** from ***j*** to ***m*** in period ***t*** (where A_{tijj}=0),

Xtij : is a 0 or 1 variable for locating ***i*** at ***j*** in period ***t,***

C_{tijkm} : is a cost of material flow between ***i*** located at ***j*** and ***k*** located at ***m*** in period ***t.***

Subject to:

$$\sum_{i=1}^{n} X_{tij} = 1; \ \ \forall t, \forall j \tag{2.2.2}$$

$$\sum_{j=1}^{n} X_{tij} = 1; \ \ \forall t, \forall i \tag{2.2.3}$$

$$Y_{tijm} = X_{(t-1)ij} \times X_{tim}; \ \forall i, \forall j, \forall t \tag{2.2.4}$$

$$X_{tij} = \{0,1\}; \ \forall i, \forall j, \forall t \tag{2.2.5}$$

With N departments, in T periods DFLP, there are $(N!)^T$ feasible combinations to evaluate, indicating the complexity of the problem. The DFLP has also been solved using different approaches including exact methods, and approximate methods to be discussed next.

2.3.1. Exact Methods

One of the two approaches presented in Rosenblatt (1986) was an exact method based on dynamic programming (***DP)*** to solve the DFLP. At each state, it solves the static layout problem to propose a number of alternative layouts. The objective is to select the sequence of layouts that minimizes the total material handling and the rearrangement costs. The method was not able to solve optimally instances with up to six departments. Due the inefficiency of exact method, a second approach was proposed to solve the problem heuristically for larger instances,

Balakrishnan et al. (1992) extended the Rosenblatt's (1986) work by considering the minimization of layout rearrangement costs subject to a budget constraint. The authors proposed two approaches: dynamic programing and ***simplex-based constrained shortest path*** (CSP) algorithms to solve to optimality the constrained DFLP. In most cases, CSP gave a better performance than DP, except when the problem size is small or the budget constraint is very tight.

Lacksonen & Enscore (1993) presented a ***QAP*** formulation to solve the dynamic FLP with the objective to minimize total traveling costs over a series of discrete time periods and to minimize the rearrangement costs of changing layouts between time periods. The authors adapted five algorithms to the dynamic formulation where the ***cutting plane*** algorithm found the best solutions to a series of realistic test problems.

Chen (1998) developed a mixed integer programming (***MIP)*** model for system reconfiguration in a dynamic cellular manufacturing environment. The objectives were to minimize the inter-cell traveling costs, machine costs and rearrangement costs for the entire planning horizon. The problem was decomposed into simpler cell formation sub-problems, each sub-problem corresponds to a different time period 't' , each is formulated as a binary-integer programming model which is solved to optimally using a commercial optimization software package.

Dunker et al. (2005) presented an algorithm combining ***dynamic programming*** and **genetic search** for solving unequal sized DFLP. The genetic algorithm creates a population of layouts and updates them for each period, while the dynamic programming evaluates the cost function for all possible sequences generated in the genetic populations.

Tian et al. (2010) proposed a ***MILP*** formulation for a dynamic FLP with unequal area departments with budget constraints. They used a laying approach to arrange departments in each planning period in order to reduce the difficulty of the continuous FLP.

Huang & Wong (2015) developed a binary ***MILP*** to deal with site layout problem with multiple construction stages, and solved it using a standard branch-and-bound algorithm. The objective function was to minimize the total cost and relocation costs for all of the involved site facilities in each construction stage, the model included safety constraints that were expressed by simple linear constraint sets.

Shafigh et al. (2015) also proposed a ***MILP*** to design the dynamic distributed layout that considers production planning and system reconfiguration over multiple time periods. This model uses the concept of resource elements to capture alternative routings for processing parts. It aims to minimize the costs associated with material handling, machine relocation, setup, inventory carrying, in-house production and subcontracting needs.

2.3.2. Heuristic methods

In this section we review the traditional approximate methods based on construction and improvement as well as optimization while reviewing in a separate section the class of meta-heuristics. Due to the inability of exact approaches for solving large-sized instances, Urban (1993) proposed a steepest-descent procedure using pairwise-interchange neighborhood to search the solution space to generate approximate solutions to DFLP. The objectives were to minimize the material handling and rearrangement costs. The proposed heuristic use the forecasted window (the number of periods in the planning horizon) for the pairwise exchange with varying lengths to find different sets of good layout.

Balakrishnan (2000) improved the solution provided by Urban's heuristic for each forecast window using a backward-pass pairwise exchange heuristic. They also combined this approach with a DP to solve the DFLP. Tests shows that this approach improves the results provided by Urban's procedure, Urban (1993).

Erel et al. (2003) proposed a three-step algorithm to solve the DFLP. Firstly, a set of viable layouts is obtained by solving the SFLP for each period. Second, a dynamic programming is used to solve the shortest path problem over these viable set layouts to obtain a final solution to the DFLP. Finally, the obtained solution was enhanced using a random descent procedure using a pairwise interchange strategy. The reported results were inaccurate due to certain typographical errors in the input data, leading the same authors to rectify the previous results with new improvement, Erel et al. (2005).

Lahmar and Benjaafar (2005) presented a multi-period model to design distributed layouts. They proposed a decomposition heuristic with an objective to minimize the sum of material flow costs and rearrangement costs over a planning horizon. The decomposition approach was based on an iterative procedure, which solves iteratively the layout problem with fixed flow, allocation and layout. The proposed method offered a robust layout design, which satisfied the necessary conditions for optimality and reduced the need for layout rearrangement between periods.

Balakrishnan and Cheng (2009) investigated the performance of the steepest descent algorithm based on pairwise exchange strategy in Urban (1993). They incorporated the rolling horizon concept and forecasting uncertainty to address DFLP. The authors demonstrated that the descent algorithm for a fixed horizon may not work well under a rolling horizon setting.

McKendall et al. (2010) developed a construction-improvement heuristic to solve the continuous DFLP. For the construction heuristic, they used a boundary search technique to deal with large-sized DFLP. The construction technique consisted of placing departments along the boundaries of already placed departments in the layout. The obtained solution was then improved using a tabu search heuristic.

In Azimi & Charmchi (2011), authors formulated a DFLP with budget constraint and developed a new efficient heuristic combining discrete event simulation and linear integer programming. In the proposed algorithm, a linear interpolation was used to change the objective function into a linear function; and the optimal solution of this model represents empirical distributions that will be used by the simulation model to determine the probability of assigning a facility to a location.

Zhao & Wallace (2014) developed a heuristic for the single-product capacitated facility layout problem to minimize the total expected material handling cost when demand is random. The problem is formulated as a two-stage stochastic program where the first stage problem is to decide on the allocation of machines/departments and the second stage problem is to assign flows between machines. The heuristic transformed the stochastic layout problem into a classical QAP by assigning weighted flows to pairs of machines. They then used a deterministic model with a maximal demand rather than the true stochastic model to approximate the solution.

2.3.3. Approximate methods

The choice of approximate meta-heuristics is made according to the characteristics of dynamic variants such as size, linearity or non-linearity complexity and objectives. The most effective approaches will be explained next.

McKendall & Liu (2012) developed three ***tabu search*** algorithms to solve the DFLP. The first one was a simple TS algorithm. The second one added to the first version diversification and intensification strategies, dynamic tabu tenure length and frequency-based memory. The third variant used a probabilistic TS algorithm. The computational results showed that quality of solutions of the second TS variant out-performed that of the other two variants and previously published meta-heuristics in the literature for the DFLP.

Baykasoglu et al. (2006) used an ***ant colony*** method to solve DFLP. They considered the unconstrained and budget constrained cases. Chen (2013) presented two Binary Coded hybrid ***ant colony*** optimization for large DFLP. These methods used encoding and decoding schemes for solution representation to improve the swapping rearrangement of facilities within their meta-heuristic framework.

Mazinani et al. (2013) proposed a ***genetic algorithm*** for the DFLP design of a flexible bay structure. The DFLP assumes that the shapes, the area of each department can change throughout the planning horizon. The objective was to find the optimal number of bays, the number of departments in each bay, and the order of departments in the layout that minimize the sum of the material handling and the rearrangement costs.

Pourvaziri et al. (2014) developed a hybrid multi-population ***genetic algorithm*** for the DFLP and a heuristic to initialize the initial populations. In the developed multi-population strategy, each population evolves independently until a pre-determined number of generations. After the independent evolution, the populations are combined to create a main population at a certain migration rate. The process continues to evolve using other genetic operators. Finally, a ***simulated annealing*** is used as a local search method until stopping criteria is met.

Madhusudanan et al., (2011) suggested a ***Simulated Annealing*** algorithm for DFLP with equal size departments. The SA algorithm was reinforced by the use of a Total Penalty Cost (TPC) function. TPC was defined as the minimum acceptable re-layout cost and used to measure the robustness of the suggested layout for the given data set.

Jolai et al., (2012) introduced a multi-objective ***particle swarm optimization*** (MOPSO) for DFLP with unequal sized departments. The multi-objective aimed to minimize the material handling and rearrangement costs and to maximize the total adjacency and distance requests. Asl & Wong (2015) investigated the unequal area DFLP and suggested a modified ***particle swarm optimization*** in order to minimize the sum of the material handling costs and sum of the rearrangement costs. They used departments and periods swapping operators to improve the solutions. Two local search methods were used to find the global best solution.

Saidi-Mehrabad & Safaei (2007) proposed an ***artificial neural network*** (ANN) approach to design the layout of dynamic cellular manufacturing systems. It was based on a mean-field theory with the objective to minimize machine, relocation and the inter-cell movement costs. A comparison between the nonlinear integer-programming model and ANN solutions was made to show the efficiency of the ANN approach.

Kia et al. (2011) modelled the layout design of dynamic cellular manufacturing systems with route selection and cell reconfiguration as an integer non-linear programming while considering single-row layout and equal area machines in each cell. The authors proposed a ***fuzzy linear programming*** approach to solve the linearized model. Fuzzy parameters such as demand fluctuation and machine capacity were presented by asymmetric trapezoidal fuzzy numbers.

Ulutas & Islier (2009) explored the ***artificial Immune Systems*** for DFLP with equal area and standardized handling equipment with identical unit costs. They proposed a ***clonal selection algorithm*** which starts with generating random strings of length = Number of facilities × Number of periods, where each substring corresponds to a layout for a certain period. Then antibodies (i.e. molecules which recognize the presence of an antigen) evolution is assured by a receptor editing process that eliminates ineffective antibodies and replace them with new ones to explore new search regions.

Finally, Moslemipour et al. (2012) give a review of intelligent methods used for designing dynamic and robust layouts in flexible manufacturing systems and present their advantages and disadvantages.

2.3.4. DFLP Summary

In a dynamically changing environment, technology and competitive world design of layouts based on static data input may not be stable, leading to the necessity to develop an adaptive layout to cater for the change in the production requirements. Adaptive layouts were modelled using dynamic formulation of the static facility layout where an optimal layout for each period is insured, and a robust formulation to select the best layout over the planning horizon were attempted.

One the most successful approaches to DFLP were the class of meta-heuristics. They are capable to deal with larger sized instances, and able to address more realistic constraints. Artificial intelligence methods are seeing some increase but at less rate then meta-heuristics. They are showing good performance and were able to compete with other approximate methods.

2.4. Works related to HealthCare

Works dealing with hospitals layout planning are been more and more addressed. The first research work dealing with hospital layout planning using operations research methods was Elshafei (1977). Using a QAP formulation, author addressed the problem of locating clinics within Cairo hospital in order to reduce patient efforts while traveling from a clinic to another. To solve it, a construction and an improvement heuristic.

Butler et al. (1992) proposed a systematic two-phase approach for hospitals layout problem and bed allocation problem. In the first stage, a QAP model is formulated to determine optimal hospital layout and allocation of beds into hospital services, and in the second stage, a simulation-optimization procedure for bed allocation is performed to evaluate the obtained results.

Yeh (2006) developed a QAP model for a case study of a hospital architectural layout and solved it using annealed neural network approach that combine the simulated annealing algorithm and the Hopfield neural network. The model aims to maximize the total layout preferences of assigning facilities on sites, to maximize the total interactive preferences between facilities and to minimize the constraints violation.

Vos et al. (2007) proposed an evaluation tool for the assessment of hospital building design aiming to assure that it supports the efficient and effective operating of care processes. For this, authors used a discrete event simulation to evaluate its flexibility and fit for operations management; and outlined some recommendation to meet future developments in operational control of building design.

Gibson (2007) used an improved approach to hospitals layout planning and design with the objective to offer an efficient delivery of patient care, to make best use of staff, equipment and facilities, to provide a safe physical environment for patients, staff and visitors, and to enable improvement in the quality and productivity of health services.

Barrett (2008) addressed the designs' weakness in Clinic of Toronto General Hospital; author used a modified Systematic Layout Planning (SLP) approach to evaluate the clinic's space usage, operation levels, flows and facility patterns. An improved and aggressive layouts were developed to change and improve current conditions and were evaluated based on a qualitative criteria.

Feyzollahi et al. (2009) formulated a QAP mathematical model for the location of hospital service units and assessment of their efficiency. Authors considered uncertainty in this formulation and proposed a robust model to it.

Motaghi et al. (2011) used three stage heuristic techniques based on qualitative objective to optimize the layout of Shafa hospital by maximizing the closeness ratio. First, the initial layout of various hospital's wards is determined. Second, a Facility Relation Chart curve is drawn for all wards of the hospital. Finally, a diamond algorithm is used to calculate the efficiency of the initial and improved states.

Arnolds et al. (2012) developed a robust optimization via simulation approach regrouping mathematical optimization, discrete event simulation (DES), and improvement heuristics to generate a robust hospital layout with stochastic patient flows through an instability analysis of different layout plans in various scenarios.

Arnolds & Nickel (2013) considered the planning of ward layouts over multiple periods using Fixed and Variable wards approach. Different objectives of fixed ward layout models are considered to minimizing either costs for installing fixed patient rooms or the number of demand violations. The variable ward layout considers movable walls that can be rearranged when needed to minimizing costs for a layout plan that satisfies demand in each period. Thus, additional costs for the walls' movement have to be minimized.

Helber et al. (2014) addressed a two-stage hierarchical facility layout planning approach for Hannover's hospital. In the first stage, authors addressed a QAP formulation to assign each department to a location and used a fix-and-optimize heuristic decomposition approach to solve it. The second stage consists on locating and wards for a given department while minimizing the consumption of resources due to transportation processes.

Lorenz et al. (2015) presented an adjacency-based method to hospital layout planning using a Newtonian gravitation model to propagate changes to a single relationship immediately to the whole space layout.

In terms of Facility Location Problems (FLoP) in HealthCare systems, this field is more developed and general then FLP for which numerous works are present in the literature (see Rahman & Smith (2000) and Daskin & Dean (2004) for a survey of these works).

Kim & Kim (2010) presented a branch and bound algorithm for determining the locations of long-term care facilities. The problem is modeled as MIP and aims to minimize the maximum load of facilities while assuming that patients are assigned to the closest facilities. The same authors in Kim & Kim (2013) presented a MIP for the public healthcare FLoP. They developed a heuristic based on lagrangian relaxation and sub-gradient optimization methods. The mathematical formulation aims to allocate the patients to the facilities with the objective of maximizing the number of served patients while considering preference of the two type of patients (e.g. low-income patients and middle- and high-income patients) for the public and private facilities.

Kim et al. (2012) treated the problem of locating public health-care facilities and allocating patients to the public facilities. For this, authors developed a MIP for this problem with the objective of minimizing the total construction cost. They also developed two types of heuristics, based on priority rules and approximate model to obtain a near optimal solution in a reasonable computation time.

Even if the research work on the hospital sector seems huge, the number of work on OT is very unpretentious. Wahed et al. (2011) presented a database application program called Operating Theatre Design Analyzer (OTDA). The program is intended to evaluate the existing OT design and provide design considerations in case of new OT design referring to international standards of healthcare facilities design.

Assem et al. (2012) applied the FLP to the OT. They improved the design of OT by generating a block layout based on a graph theoretic method called SPIRAL, which is a qualitative approach to maximize the interdepartmental adjacency of the graph layout. Lin et al. (2013) proposed a qualitative approach for designing and optimizing OTFL in hospitals. First, a systematic layout planning (SLP) is applied to design OTFL and they applied fuzzy constraint theory to evaluate the layout schemes.

2.4.1. Summary

From this literature review, we can obviously see that the layout problem in healthcare systems are not well investigated as other manufacturing systems, and need more attention, especially OTs (two works) which are the heart of any hospital and the most important piece of the puzzle for the major issues of patients' lives it involves.

Regarding to the used methods, all presented works are using either QAP or GT methods and solved the problem for small-sized structures. The two methods suffer from some weakness and limitation that we can outline as follow:

QAP:

- *Dealing only with equal sized facilities;*
- *Considering only the discrete representation;*
- *No overlapping and orientation constraints.*

GT:

- *Facilities size and shape are not considered;*
- *Dealing only with the qualitative aspect;*
- *Limited applications cases.*

Given these observations, we introduce a MIP formulation to deal with the OTLP while addressing these limitations and solving the problem to optimality for bigger instances than the existing literature.

2.5. Conclusion

In this chapter, we have presented a comprehensive literature review of the static and dynamic FLP based on a large number of literature references. It was observed that research and application works on the facility layout problem continued in several domains including more complex and realistic manufacturing and service industries.

Various approaches were employed to solve different facility layout variants with specific objective functions and constraints to meet the characteristics of the studied system. Table 2.1

gives an overall summary of different publications for the facility layout problem. The columns of the table indicate respectively the authors of a paper, the objectives, the constraints, the solution methods, the modeling approach, the formulation and computational comparison with other methods. Each row presents the main characteristics under each column's heading for each paper.

To illustrate the various notations in the table, the three well-known formulations are represented by MIP (*M*), QAP (*Q*) and GT (*G*). The specific characteristics were denoted by (*R1*) for a discrete representation; (*R2*) for a continuous representation; (*R3*) for equal-sized facilities; (*R4*) for unequal sized layout; (*R5*) for static SFLP and (*R6*) for DFLPs.

The objective functions change according to the layout design specifications: minimizing the travelling costs (*O1*); maximizing the adjacency rating (*O2*); minimizing the layout rearrangement costs (*O3*); minimizing the assignment costs (O5) and (*O6*) for minimizing travelling distances, minimizing the overall production time and maximizing the flexibility, accessibility and maintenance factors.

The choice of the solution approach can be one of the following options: Exact (*E*) algorithms and approximate methods that can be divided in traditional heuristics (*H*) such as construction and local improvement; the more recent meta-heuristics or intelligent methods (*I*). The choice depends of the size and complexity of modeling the layout problem. Exact methods are suitable for the small-sized FLPs due to large computational time requirements including simple constraints such as facilities orientation (*C1*), facilities non-overlap (*C2*) and assignment constraints (*C3*). Whereas other approximate methods could be used to solve large-sized instances with more complex constraints such as multi floor layouts with elevators locations (*C4*); single or multi-row layouts (*C5*); Input/output points locations (*C6*); problems with budget constraints (*C7*); layouts with flexible bay structure (*C8*); and other facilities with varying area over the planning horizon (*C9*)).

The meta-heuristics papers can be differentiated by the solution methodologies including: tabu search (*T*), genetic algorithm (*G*), ant colony optimization (*A*), simulated annealing (*S*) and particle swarm optimization (*P*) and their hybrids. However, the other intelligent methods for the layout problems are limited, and there are a few emerging interactive real tools to design three dimensional facility layouts.

2.6. Résumé

Dans ce chapitre, nous avons présenté un état de l'art du FLP sous environnement statique et dynamique basé sur un grand nombre de références bibliographiques. Il a été observé que la recherche et l'application du FLP continuent à avancer dans plusieurs domaines en intégrant de plus en plus des problèmes complexes et réalistes de l'industrie manufacturière.

Différentes approches ont été utilisées pour résoudre différentes variantes du FLP avec des fonctions objectifs et des contraintes spécifiques pour répondre aux caractéristiques du système étudié. Le tableau 2.1 présente un résumé global des différentes publications pour le FLP. Les colonnes du tableau indiquent respectivement les auteurs, les objectifs, les contraintes, les méthodes de résolution, l'approche de modélisation, la formulation et la comparaison avec d'autres méthodes de résolution.

Pour illustrer les diverses notations du tableau, les trois modélisations bien connues sont représentées par MIP (M), QAP (Q) et théorie des graphes (G). Les caractéristiques spécifiques au FLP ont été désignées par (R1) pour une représentation discrète ; (R2) pour une représentation continue ; (R3) pour les installations de taille égales ; (R4) pour les installations de taille inégales ; (R5) pour le SFLP et (R6) pour le DFLP.

Les fonctions objectifs changent selon les spécifications de la conception : minimiser les coûts de déplacement (O1) ; maximiser le taux de proximité (O2); minimiser les coûts de réarrangement (O3); minimiser les coûts d'affectation (O5) et (O6) pour minimiser les distances de déplacement, minimiser le temps global de production, et maximiser les facteurs de flexibilité, d'accessibilité et d'entretien.

Le choix de l'approche de la solution peut être l'une des options suivantes : les méthodes exactes (E) et les méthodes approchées qui peuvent être divisées en heuristiques traditionnelles (H) telles que celles de construction et d'amélioration ; des métaheuristiques (notations en dessous) ou des méthodes intelligentes (I). Le choix dépend de la taille et de la complexité de la modélisation du FLP. Les méthodes exactes sont adaptées à des problèmes de petite taille en raison des grandes exigences en matière de temps de calcul en incluant de simple contraintes telles que : l'orientation de installations (C1), le non-chevauchement des installations (C2) et les contraintes d'affectation (C3). Alors que, les autres méthodes approchées sont souvent utilisées pour résoudre de grandes instances, avec un temps de calcul plus réaliste et de traiter des contraintes plus complexes. Ces contraintes peuvent être : des structures en plusieurs étages avec localisation des ascenseurs (C4); structure en une ou plusieurs rangées (C5); l'emplacement des points d'entrée / sortie (C6); problèmes avec des contraintes budgétaires (C7); disposition avec cloisons (C8); et des installations avec une superficie variante de période à autre(C9)).

Les papiers sur les métaheuristiques peuvent être différenciés par les méthodes de résolution telles que : la recherche tabou (T), algorithme génétique (G), l'optimisation de colonie de fourmis (A), recuit simulé (S) et optimisation par essaim de particules (P) et leurs hybrides. Cependant, les autres méthodes intelligentes pour le FLP sont limitées, et il y a quelques outils interactifs émergents pour concevoir des espaces trois dimensionnels.

Reference	Objectives						Constraints									Representations						Formulation			Solution approaches								Comp
	O1	O2	O3	O4	O5	O6	C1	C2	C3	C4	C5	C6	C7	C8	C9	R1	R2	R3	R4	R5	R6	M	Q	G	E	H	S	A	G	T	P	I	
Al-Hakim (1992)		✓	✓				✓									✓			✓	✓				✓	✓								N
Asl & Wong (2015)	✓		✓				✓	✓									✓		✓	✓	✓	✓									✓		Y
Azimi & Charmchi (2011)	✓		✓						✓				✓			✓		✓			✓		✓			✓							Y
Balakrishnan & Cheng (2009)	✓		✓						✓							✓		✓			✓		✓			✓							Y
Balakrishnan et al. (1992)	✓		✓						✓				✓			✓		✓			✓		✓		✓								Y
Baykasoglu et al. (2006)	✓		✓						✓				✓			✓		✓			✓		✓					✓					Y
Bozer & Wang (2012)		✓					✓	✓									✓		✓	✓		✓		✓		✓							Y
Carrie et al. (1978)		✓					✓									✓		✓		✓				✓		✓							Y
Chen & Sha (2005)	✓	✓							✓							✓		✓		✓			✓			✓							N
Chen (2013)	✓		✓						✓							✓		✓			✓		✓					✓					Y
Christofides & Benavent (1989)									✓							✓		✓		✓			✓		✓								N
Dunker et al. (2005)	✓		✓				✓	✓							✓		✓		✓		✓	✓				✓							N
Erel el al. (2003)	✓		✓						✓							✓		✓			✓		✓			✓							Y
Foulds & Robinson (1978)		✓					✓											✓		✓				✓		✓							N
García-Hernández et al. (2013)	✓	✓					✓	✓						✓			✓		✓	✓		✓							✓				N
Hahn et al. (1998)	✓								✓							✓		✓		✓			✓										Y
Hathhorn et al. (2013)	✓			✓			✓	✓		✓							✓		✓	✓		✓			✓								N
Huang & Wong (2015)	✓		✓	✓		✓	✓		✓								✓		✓		✓	✓			✓								N
Jaafari et al. (2009)	✓	✓					✓	✓				✓					✓		✓	✓		✓			✓								N
Jiang & Nee (2013)	✓	✓				✓	✓	✓									✓		✓	✓		✓							✓			✓	N
Jolai el al. (2012)	✓	✓	✓			✓	✓	✓				✓					✓		✓		✓	✓				✓					✓		N
Kia el al. (2011)	✓		✓			✓		✓			✓							✓			✓	✓										✓	N
Kim & Kim (1995)	✓							✓								✓			✓	✓				✓		✓							Y
Konak et al. (2006)	✓						✓	✓						✓			✓		✓	✓		✓			✓								Y
Kothari & Ghosh (2013)						✓					✓									✓		✓				✓				✓			Y
Krishna et al. (2009)	✓	✓						✓	✓	✓										✓		✓			✓								N
Kulturel-Konak & Konak (2011)	✓						✓	✓						✓			✓		✓	✓		✓						✓					Y
Lacksonen & Enscore (1993)	✓		✓						✓							✓		✓			✓		✓		✓								N
Lahmar & Benjaafar (2005)	✓		✓					✓	✓								✓		✓		✓		✓			✓							N
Liu & Meller (2007)	✓					✓		✓									✓		✓	✓		✓				✓			✓				Y
Madhusudanan et al. (2011)	✓		✓						✓							✓		✓			✓		✓				✓						Y
Maniezzo & Colorni (1999)					✓				✓							✓		✓		✓			✓			✓		✓					Y
Marzetta & Brüngger (1999)					✓				✓							✓		✓		✓			✓		✓								Y
Mazinani et al. (2013)	✓		✓				✓	✓	✓					✓	✓		✓		✓		✓	✓							✓				Y
McKendall & Liu (2012)	✓		✓						✓							✓		✓			✓		✓			✓				✓			Y
Nyberg & Westerlund (2012)					✓				✓							✓		✓		✓		✓	✓		✓								Y
Papageorgiou & Rotstein (1998)	✓						✓	✓	✓								✓		✓	✓		✓			✓								N
Patsiatzis & Papageorgiou (2002)	✓			✓			✓	✓	✓	✓							✓		✓	✓		✓			✓								N
Pourvaziri & Naderi (2014)			✓		✓				✓												✓		✓			✓	✓		✓				Y
Şahin (2011)	✓	✓							✓							✓		✓		✓			✓				✓						Y
Saidi-Mehrabad & Safaei (2007)	✓		✓			✓			✓							✓		✓			✓	✓										✓	N
Samarghandi et al. (2010)						✓					✓					✓		✓		✓		✓									✓		Y
Seppänen & Moore (1970)		✓						✓								✓		✓		✓				✓	✓								N
Shafigh et al. (2015)	✓		✓			✓			✓								✓	✓			✓	✓			✓								N
Solimanpur & Jafari (2008)	✓							✓								✓			✓	✓		✓			✓								N
Tarkesh et al. (2009)	✓					✓			✓							✓		✓		✓			✓									✓	Y
Tian et al. (2010)	✓		✓				✓	✓					✓				✓		✓		✓	✓			✓				✓				N
Ulutas & Islier (2009)			✓		✓				✓							✓		✓			✓		✓									✓	Y
Ulutas & Kulturel-Konak (2012)	✓													✓			✓		✓	✓		✓										✓	Y
Urban (1993)	✓		✓						✓							✓		✓			✓		✓			✓							N
Zhang& Che (2014)						✓			✓							✓		✓		✓		✓			✓								N

Table 2.1. Comparison of the FLP approaches

Notes: Comp = is compared with other existing approaches? (Y = yes, N = no).

Chapter III. Static Operating Theater Layout Problem: Different Models and solutions

3.1. Introduction

The Static Operating Theater Layout problem (SOTL) is a variant of the classical Facility Layout Problem. SOTL problem is concerned with the determination of the optimal layout of placing a set of N facilities (rooms) on a plane (department). The objective is to improve the efficiency and effectiveness of the health provision process subject to a set of health-care accreditation standards, and a set of operating and logical constraints. The SOTL problem has received a little attention in the literature. Its existing SOTL models are based on graph theoretic approaches, which do not consider neither the size, the orientation and nor the shape of rooms in the modeling process. They focused just on maximizing the desirability measure on adjacency of facilities.

In this chapter, four Mixed Integer Linear Programming (MILP) models are proposed to solve the STOL problem. The STOL objectives are to minimize the interdepartmental traveling costs and to maximize the closeness among facilities. The presented unlike past models, determine the position and orientation of each facility according to the international health care accreditation standards and operating technical constraints. The proposed models are validated on several illustrative examples and solved using commercial optimization software. The following terms are used to understand the STOL problem representation in Figure 3.1:

- **Facility**: refers to a health-care service, a room, a ward, an facility or an office.
- **Section:** refers to a specific zone in the SOTL where facilities are to be placed according to some healthcare standards.
- **Department:** refers to a hospital's pavilion, a building, a module or a health-care unit where each department could contain several sections.

- **Entity**: refers to the set personnel of doctors, patients, medical staff or non-medical staff that move between facilities.

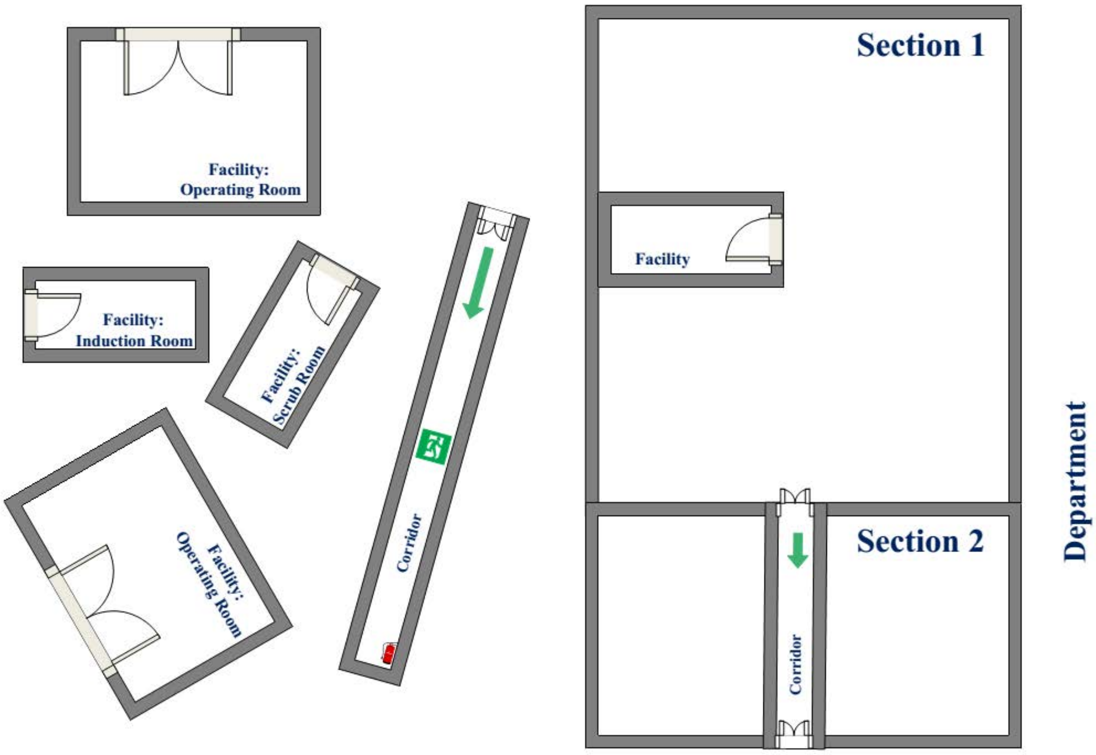


Figure 3.1. An illustration of the typical components of the SOTL problem

For the remainder of this chapter, Section 3.2 introduces the basic formulation for simple SOTL Problem, Sections 3.3, 3.4 and 3.5 present special structures of real SOTL, namely, multi-sections; multi-floors; and multi-rows. We will propose different MIP formulations to deal with each case. In each section, the proposed model, and associated experimental results are presented to demonstrate the effectiveness and efficiency of the proposed approach.

3.2. SOTL Basic Mathematical Formulation

Given a set of facilities, the dimensions and orientation of each facility and the available are of the department space, the Static Operating Theater Layout (SOTL) problem seeks to determine the optimal placement of the set of facilities within the available space subject to non-overlapping facilities on the floor plan layout while optimizing a bi-objective function. The bi-objective function includes a quantitative and qualitative measures related to the characteristics of the SOTL problem under consideration.

3.2.1. Mathematical Formulation

Assumptions

- SOTL contains a set of multi-disciplinary facilities;

- The Induction Room (IR) and the Operating Room (OR) are considered as separate facilities;
- The arsenals and other known facilities aren't considered within SOTL design;
- The facilities have rectangular shapes with unequal areas of known dimensions;
- The movement between facilities is modelled as a rectilinear centroid-to-centroid movement;
- The number of trips, the difficulty of movements, the desirable relationship values between facilities and the cost factors assigned to each entity types are given;
- The total layout area of a department is equal to the total area of facilities multiplied by a ratio 1.4 according to international standards (see Le Mandat [2001)]).

Sets

- Let N= { a_i ; i=1,2,…,n } be the set of n facilities,
- Let K= { e_k ; k=1, …,4} be the set of k entity types**.**

Parameters

α_i = Length of facility a_i

β_i = Width of facility a_i

F_{ijk}= The number of trips between facility a_i to facility a_j made by entity e_k

φ_{ijk}= The traveling difficulty between facility a_i to facility a_j made by entity e_k

σ_k= The cost factor assigned to entity e_k

X_{max}= The maximum length of department;

Y_{max}= The maximum width of a department;

R_{ij}= The desirability relationship value between facility a_i to facility a_j;

ρ_1, ρ_2= The weights for the qualitative and quantitative sub-objective functions respectively.

Decision Variables

Ω_i = { 1 if length α_i of facility a_i is parallel to x-axis (horizontal orientation)
0 otherwise

z_{ij}^{x} = { 1 if facility a_i is strictly to the right of facility a_j
0 otherwise

z_{ij}^{y}= { 1 if facility a_i is strictly above facility a_j
0 otherwise

μ_{ij}= $\begin{cases} 1 \text{ if } a_i \text{ and } a_j \text{ are fully adjacent} \\ 0 \text{ otherwise} \end{cases}$

x_i , y_i= x and y coordinates of the geometric center of gravity for facility a_i

D_{ij}= distance between facilities a_i and a_j

l_i= x-length of facility a_i along x-axis (it depends on whether α_i or β_i is parallel on x-axis)

d_i= y-length of facility a_i along y-axis (it depends on whether α_i or β_i is parallel on y-axis.

Constraints

Our formulation contains two set of constraints. First, the *formulation* constraints are to make the relationships easier to understand. Second, the *modeling* constraints are the logical relationships to generate a valid solution.

<u>*The formulation constraints*</u>**:** The values of the variables l_i and d_i depend on the orientation of facility a_i in the plane. If a_i is placed with its length α_i is parallel to the x-axis, then the orientation of a_i is said to horizontal (i.e. $\Omega_i = 1$) and l_i = α_i, (the x-length) and d_i = β_i $(the\ width)$. Otherwise a_i is vertical and β_i is parallel to y-axis, leading to l_i = β_i (the y-length), and l_i =α_i $(the\ width)$. Figure 3.2 illustrates the two different orientations of facilities a_i and a_j and their associated lengths; whereas Equations (3.2.1 and 3.2.2) expresses the length relationships.

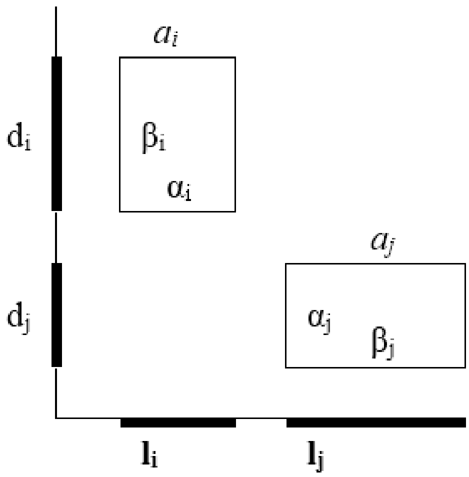


Figure 3.2. ***l_i and d_i according to layout's orientation***

$$l_i = \alpha_i \Omega_i + \beta_i (1 - \Omega_i) \qquad \forall i \in N \qquad (3.2.1)$$

$$d_i = \alpha_i (1 - \Omega_i) + \beta_i \Omega_i \qquad \forall i \in N \qquad (3.2.2)$$

<u>*The logical constraints:*</u>

a) No occupancy of the same physical location by any two facilities: The no-occupancy constraints are expressed by the following two equations:

$$|x_i - x_j| \geq \frac{l_i + l_j}{2} \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.3}$$

$$|y_i - y_j| \geq \frac{d_i + d_j}{2} \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.4}$$

The two constraints ensure that no two facilities a_i and a_j should occupy the same physical location. The two facilities should be separated either on the x-direction or on the y-direction (see Figure 3.3). Since the two restrictions involve an absolute relationship, they can be modeled using as a mixed integer linear constraints using equations (3.2.5) to represent the case 1 and (3.2.6) to represent case 2 in Figure 3.3:

$$x_i - x_j + M\left(1 - z_{ij}^x\right) \geq \frac{l_i + l_j}{2} \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.5}$$

$$y_i - y_j + M\left(1 - z_{ij}^y\right) \geq \frac{d_i + d_j}{2} \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.6}$$

Where: M is a suitable upper bound on the distance between two facilities (could take the value of the length of the department)

b) No overlapping between pairs of facilities: This restriction can be enforced by having at least one of the logical variables associated with constraints (3.2.5) or (3.2.6) set to one as follows:

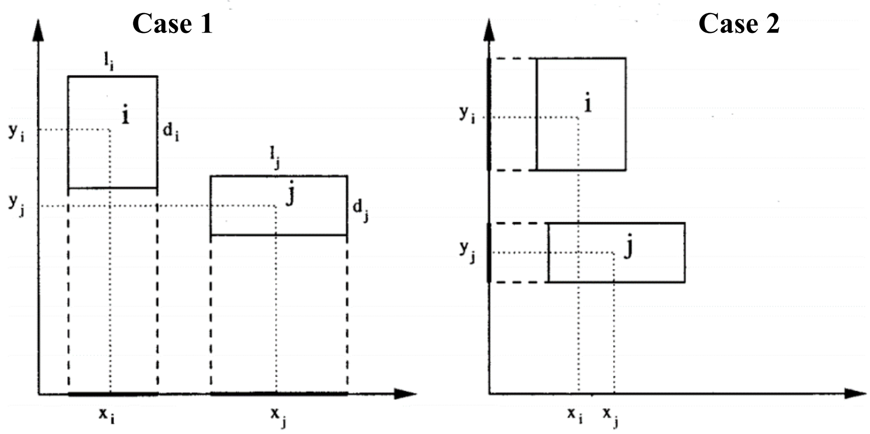


Figure 3.3. Non-Overlapping constraints

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \geq 1 \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.7}$$

$$z_{ij}^x + z_{ji}^x \leq 1 \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.8}$$

$$z_{ij}^{y} + z_{ji}^{y} \leq 1 \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.9}$$

c) Boundary constraints: the coordinates of all facilities determined by their center of gravities must lie within the corners (0, 0) and (X_{max} , Y_{max}) of the layout space of the department. They are expressed as follows:

$$x_i \geq \frac{l_i}{2} \quad \forall i \in \mathrm{N} \tag{3.2.10}$$

$$y_i \geq \frac{d_i}{2} \quad \forall i \in \mathrm{N} \tag{3.2.11}$$

$$x_i + \frac{l_i}{2} \leq X_{max} \; \forall i \in \mathrm{N} \tag{3.2.12}$$

$$y_i + \frac{d_i}{2} \leq Y_{max} \; \forall i \in \mathrm{N} \tag{3.2.13}$$

d) Distance between any two facilities will be calculated using the Manhattan distance expressed in equations (3.2.14):

$$D_{ij} = |x_i - x_j| + |y_i - y_j| \qquad \forall\, i = 1, \dots, n-1;\; j = i+1, \dots, n \tag{3.2.14}$$

e) Since the coordinates of each facility placement must be determined, and then the Manhattan distance between facilities are computed. This process would make our mathematical formulation as a mixed integer non-linear continuous model. Thus, the absolute terms in the constraints (3.2.15) to (3.2.18) need to be linearized using the following relationships:

$$D_{ij} \geq (x_i - x_j) + (y_i - y_j) \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.15}$$

$$D_{ij} \geq (x_i - x_j) + (y_j - y_i) \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.16}$$

$$D_{ij} \geq (x_j - x_i) + (y_i - y_j) \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.17}$$

$$D_{ij} \geq (x_j - x_i) + (y_j - y_i) \quad \forall\, i, j \neq i \in \mathrm{N} \tag{3.2.18}$$

Objective functions

The main objective of the SOTLP is to provide the best placement of facilities within the available space. In the literature, different objectives are found and the choice of one over the other depends on the application domain. In this work, a multi-objective function as a weighted sum of two sub-objectives is computed.

The first sub-objective is a quantitative objective, which aims to minimize the total travel costs. It is proportional to the rectilinear distance, travel frequency, and trip difficulty rating. The baseline travel cost is computed using equation (3.2.19):

$$\boldsymbol{Min\ F_1} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{4} \boldsymbol{F_{ijk} D_{ij} (\varphi_{ijk} * \sigma_k)} \tag{3.2.19}$$

Where:

$\boldsymbol{F_{ijk}}$, is the Travel Frequency Attribute. It represents the number of trips made from facility a_i to facility a_j by an entity of type e_k.

$\boldsymbol{\varphi_{ijk}}$, is the trip difficulty rating. It represents the difficulty of an entity e_k to move from facility a_i to facility a_j (see Table 3.1). It is proportional to the required resource and the required effort.

$\boldsymbol{\sigma_k}$, is the baseline travel cost. It represents the cost factor assigned to entity e_k based on the type of human resources involved. Based on Roanne's hospital, the following relationships and data are provided:

$$\boldsymbol{\sigma_{doctor}} > \boldsymbol{\sigma_{medical\ staff}} > \boldsymbol{\sigma_{patient}} > \boldsymbol{\sigma_{non-medical\ staff}}.$$

Type	Entities	Rating	Movement Description
1	Patients	1.5	Movement requires some resources
2	Doctors	1	Completely independent
3	Medical staff	1.5	Movement done with a patient or some equipment
		1	Movement done freely
4	Non-Medical Staff	1.5	Movement done with a patient or some equipment
		1	Movement done freely

Table 3.1. Trip ratings based on movement description and entity types

The second sub-function is a qualitative objective, which aims to maximize the desired closeness-rating factor based on international standards. The closeness-rating chart is a qualitative grid of the desired closeness between any two facilities. The rate determines the strength of the closeness using AEIOUX abbreviation, which is explained as follows: Necessary (A), very important (E), important (I), ordinary importance (O), unimportant (U) and undesirable (X). An example of rating is presented in Figure 3.4, where OR denotes the Operating Room and PACU denotes the Post Anesthesia Care Unit. The (AEIOUX) rates are subjectively defined in Muther & Wheeler (1994) to reflect on the following work conditions:

a) Whether similar equipment or facilities are used, or similar work performed;

b) Sharing the same personnel, records, and communication;

c) Traveling frequency; etc.

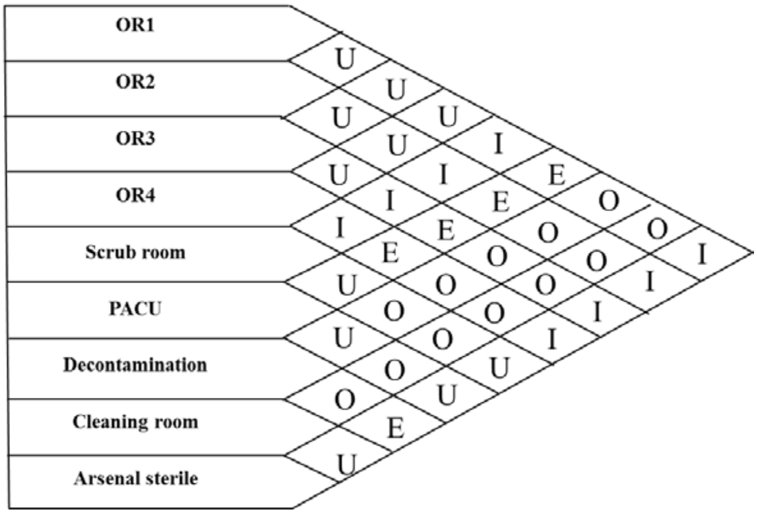


Figure 3.4. Example of AEIOUX rating of an SOTL design

The sub-objective qualitative function is expressed using Equation (3.2.20):

$$\boldsymbol{max\ F_2 = \sum_{i=1}^{N} \sum_{j=1}^{N} R_{ij}\mu_{ij}} \tag{3.2.20}$$

Where:

$\mathbf{R_{ij}}$: is the proximity relationship, which expresses the *desirability* of locating adjacent facilities next to each other, where a strong positive relationship indicates a proximity rank of A. The AEIOUX rates are associated to values, equal to A=16, E=8, I=4, O=2, U=0 and X=-2, respectively and are defined according to the international standards in terms of hygiene.

$\boldsymbol{\mu_{ij}}$: is the adjacency coefficient expressing the *proximity* between two facilities according to the distance separating them. The adjacency coefficients are defined as:

- $\boldsymbol{\mu_{ij}} = 1$ if two facilities are fully adjacent, i.e. the two facilities are facing directly each other or are sharing the same wall.

- $\boldsymbol{\mu_{ij}} = 0$ if two facilities are non-adjacent: If the facilities do not share any point or when they cannot be seen together in the same area (septic/aseptic area).

The final objective function is a weighted sum of the two sub-objective functions expressed using Equation (3.2.21):

$$\boldsymbol{Min: F = \rho_1 F_1 - \rho_2 F_2}$$

$$\boldsymbol{Min\ F = \rho_1 \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{4} F_{ijk} D_{ij}(\varphi_{ijk} * \sigma_k) - \rho_2 \sum_{i=1}^{n} \sum_{j=1}^{n} R_{ij} \mu_{ij}} \tag{3.2.21}$$

The choice of weights for ρ_1 and ρ_2 would influence the final layout solution in such a way that the best layout using one set of weights in (3.2.19) would be probably be different from the best layout from different set of weights. Meller & Gau (1996a) discussed how different values of ρ_1, ρ_2 can be set and whether their exact values are critical or not.

Furthermore, Singh & Singh (2010) proposed four different methods to calculate relative weights for each objective measure, namely, Mean Weight Method (MWM), Geometric Mean Weight Method (GMWM), Standard Deviation Weight Method (SDWM), and CRitical Importance Through Inter-criteria Correlation Method (CRITICM). The resulted two weights were $\boldsymbol{\rho_1} = 0{,}5899$ and $\boldsymbol{\rho_2} = 0{,}4101$ for MWM, $\boldsymbol{\rho_1} = 0{,}5036$ and $\boldsymbol{\rho_2} = 0{,}4964$ for GMWM, $\boldsymbol{\rho_1} = 0{,}5051$ and $\boldsymbol{\rho_2} = 0{,}4949$ for SDWM and $\boldsymbol{\rho_1} = 0{,}5036$ and $\boldsymbol{\rho_2} = 0{,}4964$ for CRITICM. As observed, there is no large difference between the two weights when solving the multi-objective FLP.

Similarly, Sha & Chen (2001) proposed to solve a multi-objective FLP using the probability of superiority as a measure of solution quality for the determination of the probability that one layout is better than the others while trying different values of weights ρ_1 and ρ_2. We observe that the best solution was obtained when $\boldsymbol{\rho_1} = 0{,}6$ and $\boldsymbol{\rho_2} = 0{,}4$ or when $\boldsymbol{\rho_1} = 0{,}5$ and $\boldsymbol{\rho_2} = 0{,}5$.

We have not been content to simply observe the conclusions of these works, but we tried to solve our multi-objective problem under different values of weights to observe their impact on the quality of final solution. As result, international standards were respected only when $\boldsymbol{\rho_1}$ and $\boldsymbol{\rho_2}$ were close together to keep balance between the quantitative and the qualitative objectives. Thus, in the remainder of this dissertation, the values of the two weights are fixed to $\boldsymbol{\rho_1} = 0{,}5$ and $\boldsymbol{\rho_2} = 0{,}5$.

3.2.2. Computational Experience

In our computational experience, two SOTL models are considered: the discrete and continuous representations. In the discrete model, the coordinates of center of gravities for facilities are restricted to take the integer values on the rectangular plane defined by the two extreme corner points (0, 0) and (Xmax ,Ymax), whereas in the continuous model the such coordinates are allowed to take real-values within the boundary of the plane. For each model, two different instances were generated for testing. The generated instances respect the international standards in terms of security and proximities between critical facilities.

ILOG CPLEX 12.2 software was used to solve the model using Windows 7 platform, on Personal computer with the following specifics: Intel5® Core ™ i5-2410M CPU@ 2.30GHz and 6GB of RAM. For all instances: $\rho_1 = \rho_2 = 0.5$, M=100, $\sigma_{doctor} = 80,\ \sigma_{medical\ staff} = 60,\ \sigma_{patient} = 40\ and\ \sigma_{non-medical\ staff} = 20$ are used.

i) Discrete Model Instances

The discrete model instances were generated by restricting (x_i, y_i) to take integer values between the two corner points. Two instances were generated with the following characteristics: Instance 1 is composed of nine facilities and instance 2 is composed of eleven facilities. Since the number of facilities is limited, we select just the necessary and the most important ones to the OT functioning.

Instance 1 is composed of nine Facilities to be placed within the two corner points with (X_{min}= 0, Y_{min}= 0) and (X_{max}= 19, Y_{max}=19). The characteristics of the facilities and their dimensions are reported in Table 3.2. Further, the following scenario is considered: OR1 and OR2 facilities share the same induction and scrub room. The optimal solution and orientation of facilities for this instance are shown in Table 3.3. The total processing time to obtain the optimal solution was equal to 13.63 CPU seconds.

Instance 2 is composed of 11 facilities to be placed within the two corner points with (X_{min}= 0, Y_{min}= 0) and (X_{max}= 20, Y_{max}= 26). In this instance, large facilities and more relationships are considered.

The OR1 and OR2 share the induction room1 and the OR3 is served by the induction room2, while the scrub room 1 is shared by the three OR as shown in Table 3.4. Table 3.5 shows the optimal solution obtained after 9604.47 CPU seconds.

Facilities	Dimensions	
	αi	βi
Induction	4.00	3.00
OR1	10.00	5.00
OR2	7.00	6.00
Decontamination	4.00	3.00
PACU	10.00	6.00
Cleaning room	3.00	2.00
Scrub room	3.00	2.00
Semi- Protected corridor	19.00	2.00
Protected corridor	19.00	2.00

Table 3.2. Dimensions of named facilities in instance 1

Facilities	Orientation		Optimal location	
	li	Di	xi	yi
Induction	4.00	3.00	11.00	12.00
OR1	10.00	5.00	8.00	16.00
OR2	6.00	7.00	16.00	14.00
Decontamination	4.00	3.00	5.00	12.00
PACU	10.00	6.00	9.00	3.00
Cleaning room	2.00	3.00	3.00	4.00
Scrub room	2.00	3.00	8.00	12.00
Semi- Protected corridor	19.00	2.00	9.00	7.00
Protected corridor	19.00	2.00	9.00	9.00

Table 3.3. Optimal Orientation and Location of facilities in instance 1

Facilities	Dimensions	
	αi	βi
Induction1	4.00	3.00
Induction2	4.00	3.00
OR1	10.00	5.00
OR2	10.00	5.00
OR3	7.00	6.00
Decontamination	4.00	2.00
PACU	12.00	7.00
Cleaning room	3.00	2.00
Scrub room1	3.00	2.00
Semi- Protected corridor	19.00	2.00
Protected corridor	19.00	2.00

Table 3.4. Dimensions of named facilities in instance 2

Facilities	Orientation		Optimal location	
	li	di	xi	yi
Induction1	3.00	4.00	13.00	12.00
Induction2	3.00	4.00	10.00	22.00
OR1	5.00	10.00	11.00	5.00
OR2	5.00	10.00	17.00	9.00
OR3	7.00	6.00	15.00	21.00
Decontamination	4.00	2.00	9.00	13.00
PACU	7.00	12.00	5.00	6.00
Cleaning room	3.00	2.00	10.00	13.00
Scrub room1	3.00	2.00	10.00	11.00
Semi- Protected corridor	19.00	2.00	10.00	15.00
Protected corridor	19.00	2.00	10.00	17.00

Table 3.5. Optimal orientation and location of facilities in instance 2

As refer to international standards, it was observed that the discrete model does not provide satisfactory results since some facilities could not be placed in the appropriate section. This result is due to the fact that the locations of the centroid of facilities are restricted to take integer values. To consider the impact of this restriction, the centers are relaxed to take instead real-floating values.

ii) Continuous Model Instances

The continuous models allows the centers of gravities to be placed anywhere on the plane. Two new instances of the discrete model are considered for the continuous model except of the integer restrictions are relaxed. The corresponding instances are called instance 3 of nine facilities and instance 4 of eleven facilities.

Facilities	Dimensions	
	αi	βi
Induction	4.60	3.20
OR1	8.20	7.20
OR2	8.20	6.40
Decontamination	4.20	3.20
PACU	12.00	6.00
Cleaning room	3.00	4.20
Scrub room	4.60	2.60
Semi- Protected corridor	18.20	2.40
Protected corridor	18.20	2.40

Table 3.6. Dimensions of the named facilities in instance 3

Instance 3: Table 3.6 shows the dimensions of the nine facilities Whereas Table 3.7 shows the optimal layout for this instance. The required computation time to obtain the optimal solution was equal to 3.60 CPU second. Figure 3.5 shows the optimal drawing for the layout of this instance.

Facilities	Orientation		Optimal location	
	li	di	xi	yi
Induction	4.60	3.20	8.70	4.00
OR1	7.20	8.20	14.60	4.10
OR2	6.40	8.20	3.20	4.10
Decontamination	3.20	4.20	13.60	15.10
PACU	12.00	6.00	6.00	16.00
Cleaning room	3.00	4.20	16.70	15.10
Scrub room	4.60	2.60	8.70	6.90
Semi- Protected corridor	18.20	2.40	9.10	11.80
Protected corridor	18.20	2.40	9.10	9.40

Table 3.7. Optimal orientation and location of instance 3

Instance 4: The OR1 and OR2 share induction room1 while induction room2 serves OR3. Table 3.8 gives dimensions of named facilities, while Table 3.9 shows the details of the optimal solution obtained after 114.21 sec (See Figure 3.6).

Facilities	Dimensions	
	αi	βi
Induction1	6.40	2.60
Induction2	5.20	2.80
OR1	8.80	5.00
OR2	5.40	5.40
OR3	10.60	6.00
Decontamination	2.40	4.00
PACU	11.80	8.60
Cleaning room	4.60	3.00
Scrub room1	2.60	4.60
Semi- Protected corridor	17.60	2.40
Protected corridor	17.60	2.40

Table 3.8. Dimensions of the named facilities of instance 4

The first advantage here is that facilities have more liberty to occupy any desired position in the layout, which eliminate the restrictions of the discrete representation. The second advantage is that the obtained layouts in this continuous representation respect the international standards in terms of separating facilities according to exigencies of hygiene and security.

Using the continuous representation, we solved the OTL using only the quantitative objective function. The resulting layout respected also the facilities' partition, but it contained more adjacent facilities to the removal of the adjacency qualitative sub-objective. The computation time also increases for the same instance; we obtained an optimal solution with eleven facilities in 241.85 sec. Thus, the use of weighted objective function proved necessary to obtain best results and meet both qualitative and quantitative objectives at reasonable computation time.

Facilities	Orientation		optimal location	
	li	di	xi	yi
Induction1	6.40	2.60	1.90	14.70
Induction2	5.20	2.80	7.00	17.40
OR1	5.00	8.80	2.50	20.40
OR2	5.40	5.40	12.90	16.10
OR3	10.60	6.00	8.30	21.80
Decontamination	2.40	4.00	15.00	6.60
PACU	13.80	8.60	7.90	4.30
Cleaning room	3.00	4.60	15.30	2.30
Scrub room1	4.60	2.60	7.90	14.70
Semi- Protected corridor	17.60	2.40	10.20	9.80
Protected corridor	17.60	2.40	10.20	12.20

Table 3.9. Optimal orientation and location of Instance 4

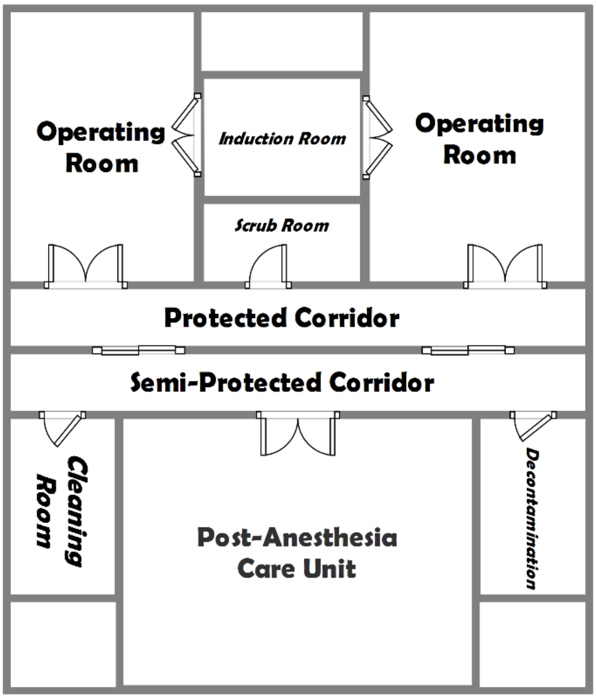


Figure 3.5. Optimal Layout of Instance 3

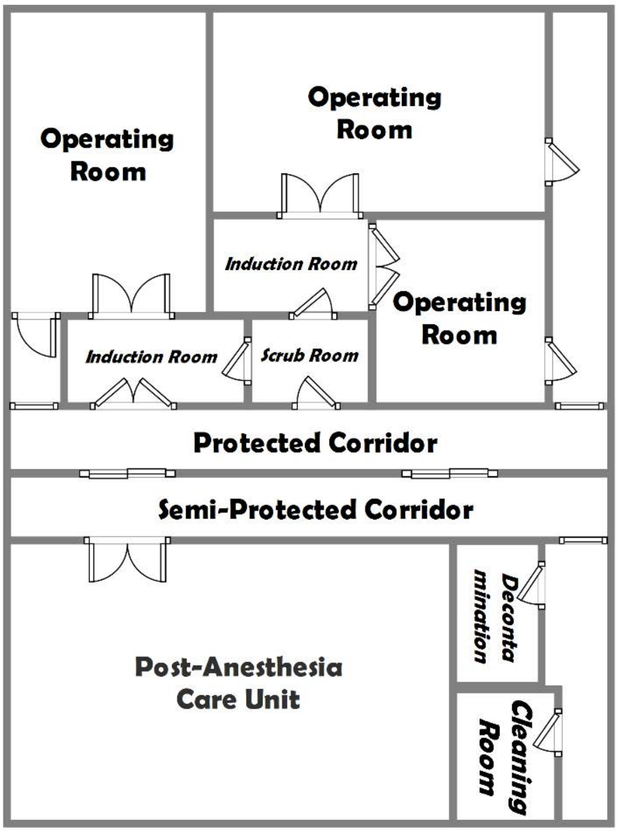


Figure 3.6. Optimal Layout of Instance 4

In another hand, we tried to solve the discrete representation by relaxing the integer value of facilities location (i.e. x_i and y_i) to compare the obtained solution with the continuous representation's ones. Optimal solution was reached also but a higher CPU time was observed (i.e. 352.90 sec) and final layout does not necessarily respect the international standards in terms of locating sensitive facilities in the appropriate zone. Thus, the continuous representation proves to be more effective than the discrete representation both in terms of CPU time and in terms of solution's quality.

3.2.3. Summary

The presented model aims to determine the optimal layout for the SOTLP in two-dimensional space under discrete and continuous representations. Mixed Integer Programming models were developed to determine the optimal location and orientation of each facility in the final layout while minimizing the total traveling cost and maximizing the desirable closeness rating. Few different real-life scenarios were considered such as the separation of restricted and semi-restricted area, distinction between OT flows, different cost and difficulty to each entity, and orientation of facilities. The applicability of the proposed model was demonstrated on four typical instances under discrete and continuous representations, for each the model gives optimal solutions respectively for nine and eleven facilities.

3.3. Multi-Section SOTLP Mixed Integer Models

In the previous section, we considered the case of Operating Theater consisting of one section, however, the safety and hygiene consideration necessitates an organization to have layout design consisting of well-defined multi-section layout to prevent contamination, to avoid accident, and to facilitate the movement of material and different actors of the operating theater. The new variant is formulated with the objective to determine the locations of both the sections and the facilities within each section that minimizes the total traveling costs and to maximize the total adjacency rating. The main idea of the multi-section SOTL formulation is inspired by analogy to the multi-floor layout problem [Krishna et al. (2009)]. In the analogy, floors are represented by adjacent sections, and elevators to move between floors are replaced by corridors to move between sections.

3.3.1. Mathematical Formulation

Given a set of facilities, a set of corridors and a number of sections, the multi-section OTLP consists on assigning facilities to sections and using corridors to connect facilities within a section and across sections in the whole layout while optimizing the objective functions.

Assumptions

For the multi-section SOTLP have the same assumption of section 3.2 in addition to the following ones:

- The orientation of the final layout is a given parameter;
- The maximum number of corridors and their orientations are given. In the solution, corridors are used to travel between two facilities in the same section, and to transit from one section to another;
- The location of a corridor is a decision variable;
- The corridors are aligned according to the layout orientation to insure the entrance to the OT unit, the movements between sections and the exit from the OT unit (see Figure 3.7);
- No relationships of facilities with the outside world are allowed;
- A single facility cannot be split among multiple sections;
- The transition between sections can only occur through corridors;
- The shape and area of the facilities are given. For each facility, the associated decision variables are its location and horizontal (or vertical) orientation.

Sets

- Let N= { a_i ; i=1,2,…,n } be the set of n facilities,

- Let K= { e_k ; k=1, …,4} be the set of k entity types: doctor, patient, medical staff or non-medical staff.
- Let S= {ε_s ; s=1, 2,…,t } be the set of sections: outer, restricted, aseptic, disposal….
- Let C= {θ_c; c=1, 2,…,r } be the set of corridors: clean, public, etc.
- Let U_i be the set of a single element denoting the section to which facility a_i is belonging.
- Let U_c be a set of single element denoting the section to which corridor θ_c is belonging.

Parameters

α_i :		Length of facility a_i
β_i :		Width of facility a_i
l_c, d_c:		Dimensions of corridor θ_c
Ω_s	{	1 if length (Xmax) of the layout is parallel to x-axis (horizontal orientation) 0 otherwise
F_{ijk}:		Number of trips between facility a_i to facility a_j made by an entity type e_k
φ_{ijk}:		Moving difficulty between facility a_i to facility a_j made by an entity type e_k
σ_k:		Cost factor assigned to entity e_k
X_{max}:		Maximum length of the department ;
Y_{max}:		Maximum width of the department ;
R_{ij}:		Desirable relationship value between facility a_i to facility a_j;
ρ_1, ρ_2:		Weights for each sub-objective function,
$Xl_s, Xr_s,$ Yt_s, Yb_s		X and Y boundary coordinates of section ε_s

Decision Variables

Ω_i	{	1 if length α_i of facility a_i is parallel to x-axis 0 otherwise
μ_{ij}	{	1 if a_i and a_j are fully adjacent 0 otherwise
V_{ijc}	{	1 if traffic between facilities a_i and a_j travels through corridor θ_c 0 otherwise
t_{ij}	{	1 if facility a_i and a_j are assigned to the same section 0 otherwise
t_{ic}	{	1 if facility a_i and corridor θ_c are assigned to the same section

0 otherwise

v_{is}	{	1 if facility a_i is assigned to section ε_s 0 otherwise
v_{cs}	{	1 if corridor θ_c is assigned to section ε_s 0 otherwise
z_{ij}^x	{	1 if facility a_i is strictly to the right of facility a_j 0 otherwise
z_{ij}^y	{	1 if facility a_i is strictly above facility a_j 0 otherwise
o_{ic}^x	{	1 if facility a_i is strictly to the right of corridor θ_c 0 otherwise
o_{ic}^y	{	1 if facility a_i is strictly above corridor θ_c 0 otherwise
x_i , y_i:		x and y coordinates of the geometric center of gravity facility a_i
x_c ,y_c :		coordinates of the geometrical center of corridor θ_c
l_i:		x-length of facility a_i depending on whether α_i or β_i is parallel on x-axis
d_i:		y-length of facility a_i depending on whether α_i or β_i is parallel on y-axis
Xp_{ij}:		x-distance between facility a_i and a_j
Yp_{ij}:		y-distance between facility a_i and a_j

Constraints

a) Orientation constraints (facilities and corridors)

These constraints are the same as those in (3.2.1) and (3.2.2) :

$$l_i = \alpha_i\Omega_i + \beta_i(1 - \Omega_i) \qquad \forall i \qquad (3.3.1)$$

$$d_i = \alpha_i + \beta_i - l_i \qquad \forall i \qquad (3.3.2)$$

b) Sections constraints

Constraints (3.3.3) ensure that each facility is assigned to only one section. Constraint (3.3.4) is used to obtain the value of t_{ij}: If two facilities a_i and a_j are allocated to the same section ($v_{is} = v_{js} = 1$), then constraint (3.3.4) will assure that t_{ij}=1. Else, if facilities a_i and a_j are allocated to different sections, then constraint will ensure that t_{ij}=0.

$$\sum_{s=1}^{S} v_{is} = 1 \qquad \forall i \tag{3.3.3}$$

$$t_{ij} = \left| v_{is} + v_{js} - 1 \right| \ \forall\, i, j \neq i, s \tag{3.3.4}$$

c) Corridors constraints

Constraint (3.3.5) ensures that the routing of flow between two facilities is through the corridors in the case when they are not in the same section.

$$\sum_{c=1}^{C} V_{ijc} = 1 - t_{ij} \quad \forall\, i, j \neq i \tag{3.3.5}$$

The corridors must have common boundaries to route flows between the two sections, and common boundaries with the entrance or the exit of the section to facilitate the communication with the outside. Thus, according to layout orientation, constraint (3.3.6) assures the adjacency between corridors, while constraint (3.3.7) forces the corridors to be horizontally or vertically aligned to insure travel between sections. In Figure 3.7, case 1 presents an example when $\boldsymbol{\Omega}_s =$ 0 while case 2 and corridors are aligned vertically. Constraint (3.3.8) makes sure that each corridor is assigned to only one section.

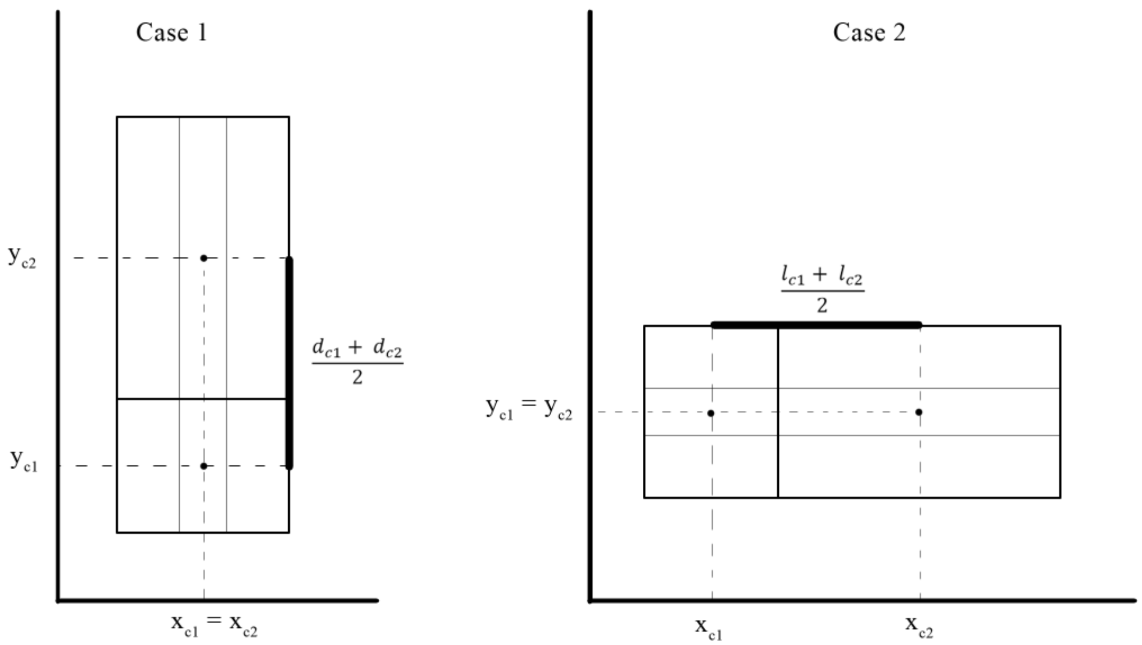


Figure 3.7. Corridors' coordinates according to layout's orientation

$$(1 - \Omega_s)|y_{c1} - y_{c2}| + \Omega_s|x_{c1} - x_{c2}| = (1 - \Omega_s)\left(\frac{d_{c1} + d_{c2}}{2}\right) + \Omega_s\left(\frac{l_{c1} + l_{c2}}{2}\right) \quad \forall\, c1, c2 \in C \tag{3.3.6}$$

$$(1-\Omega_s)x_{c2} + \Omega_s . y_{c2} = (1-\Omega_s)x_{c1} + \Omega_s . y_{c1} \quad \forall\, c1, c2 \in C \tag{3.3.7}$$

$$\sum_{s=1}^{S} v_{cs} = 1 \qquad \forall\, c \tag{3.3.8}$$

d) Facilities non-overlapping constraints

As in the previous formulation, to avoid overlapping facilities we use the mixed integer constraints (3.3.9) to (3.3.13):

$$x_i - x_j + X_{max}\left(1 - z_{ij}^x\right) \geq \frac{l_i + l_j}{2} \quad \forall\, i, j \neq i \tag{3.3.9}$$

$$y_i - y_j + Y_{max}\left(1 - z_{ij}^y\right) \geq \frac{d_i + d_j}{2} \quad \forall\, i, j \neq i \tag{3.3.10}$$

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \geq 1 \quad \forall\, i, j \neq i \tag{3.3.11}$$

$$z_{ij}^x + z_{ji}^x \leq 1 \quad \forall\, i, j \neq i \tag{3.3.12}$$

$$z_{ij}^y + z_{ji}^y \leq 1 \quad \forall\, i, j \neq i \tag{3.3.13}$$

e) Bounding constraints

Constraints (3.3.14) through (3.3.17) indicate that facilities have to be allocated within the appropriate sections' space defined by the corners (0, 0) and (Xr_s, Yt_s), while (3.3.18) through (3.3.21) insure the same bounding for corridors.

$$x_i + \frac{l_i}{2} \leq Xr_s \quad \forall\, s, i \text{ where } s = U_i \tag{3.3.14}$$

$$y_i + \frac{d_i}{2} \leq Yt_s \quad \forall\, s, i \text{ where } s = U_i \tag{3.3.15}$$

$$x_i - \frac{l_i}{2} \geq Xl_s \quad \forall\, s, i \text{ where } s = U_i \tag{3.3.16}$$

$$y_i - \frac{d_i}{2} \geq Yb_s \quad \forall\, s, i \text{ where } s = U_i \tag{3.3.17}$$

$$x_c + \frac{l_c}{2} \leq Xr_s \quad \forall\, s, c \text{ where } s = U_c \tag{3.3.18}$$

$$y_c + \frac{d_c}{2} \leq Yt_s \quad \forall\, s, c \text{ where } s = U_c \tag{3.3.19}$$

$$x_c - \frac{l_c}{2} \geq Xl_s \quad \forall\, s, c \text{ where } s = U_c \tag{3.3.20}$$

$$y_c - \frac{d_c}{2} \geq Yb_s \quad \forall\, s, c \text{ where } s = U_c \tag{3.3.21}$$

f) Corridors & facilities non-overlapping constraints

Constraints (3.3.22) to (3.3.26) ensure no-overlap between a facility a_i and a corridor θ_c if they occupy the same section; or ensure they are in different sections.

$$\left(x_c + \frac{l_c}{2}\right) \leq x_i + \frac{l_i}{2} + X_{max}(1 - o_{ic}^{x}) \quad \forall\, i, c \text{ where } U_i = U_c \tag{3.3.22}$$

$$\left(x_c - \frac{l_c}{2}\right) + X_{max}(1 - o_{ci}^{x}) \geq x_i + \frac{l_i}{2} \quad \forall\, i, c \text{ where } U_i = U_c \tag{3.3.23}$$

$$\left(y_c + \frac{d_c}{2}\right) \leq y_i + \frac{d_i}{2} + Y_{max}(1 - o_{ic}^{y}) \quad \forall\, i, c \text{ where } U_i = U_c \tag{3.3.24}$$

$$\left(y_c - \frac{d_c}{2}\right) + Y_{max}(1 - o_{ci}^{y}) \geq y_i + \frac{y_i}{2} \quad \forall\, i, c \text{ where } U_i = U_c \tag{3.3.25}$$

$$o_{ic}^{x} + o_{ci}^{x} + o_{ic}^{y} + o_{ci}^{y} \geq 1 \quad \forall\, i, c \text{ where } U_i = U_c \tag{3.3.26}$$

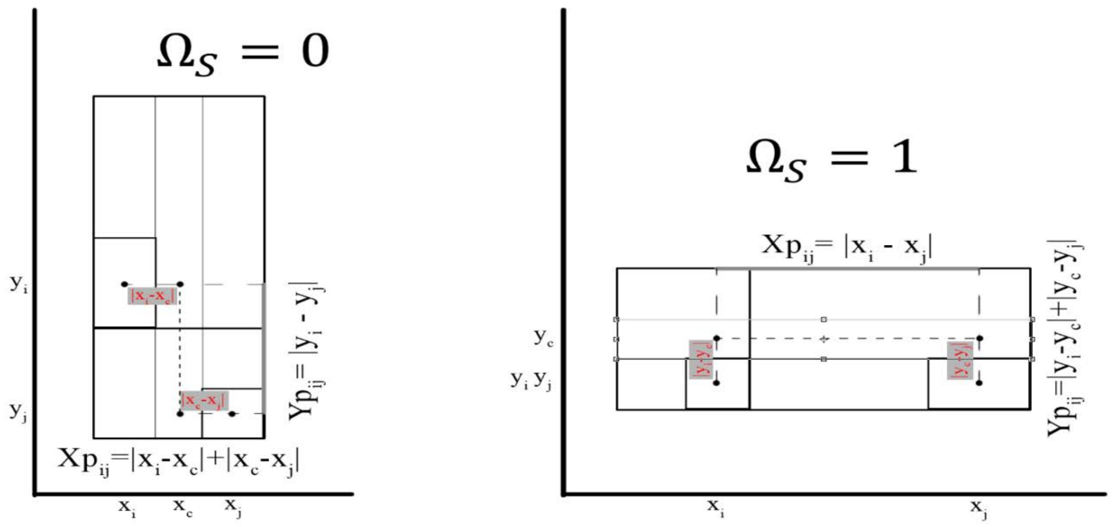
g) Distance constraints

For distance, we calculate the travel between any two facilities by going through corridors using the Manhattan norm with constraints (3.3.27) and (3.3.28) according to layout's orientation (see Figure 3.8).

$$Xp_{ij} = (1 - \Omega_s)\left(|x_c - x_i| + |x_j - x_c|\right) + \Omega_s|x_i - x_j| \quad \forall\, i, j \neq i, c \tag{3.3.27}$$

$$Yp_{ij} = (1 - \Omega_s)|y_i - y_j| + \Omega_s(|y_c - y_i| + |y_j - y_c|) \quad \forall\, i, j \neq i, c \tag{3.3.28}$$

$$D_{ij} = Xp_{ij} + Yp_{ij} \tag{3.3.29}$$



(a) ***(b)***

Figure 3.8. Distance measuring according to layout's orientation

h) Constraints linearization

To linearize constraint (3.3.4), we replace it with constraints (3.3.30) to (3.3.32) to obtain the value of t_{ij}. If two facilities a_i and a_j are allocated to the same section ($v_{is} = v_{js} = 1$), then constraint (3.3.30) will have $t_{ij}=1$, while constraints (3.3.31) and (3.3.32) remain inactive. Else, if facilities a_i and a_j are allocated to different sections, then constraints (3.3.30), (3.3.31) and (3.3.32) will ensure that $t_{ij}=0$.

$$t_{ij} \geq v_{is} + v_{js} - 1 \;\; \forall\, i, j \neq i, s \tag{3.3.30}$$

$$t_{ij} \leq 1 - v_{is} + v_{js} \;\; \forall\, i, j \neq i, s \tag{3.3.31}$$

$$t_{ij} \leq 1 + v_{is} - v_{js} \;\; \forall\, i, j \neq i, s \tag{3.3.32}$$

To linearize constraints (3.3.6), (3.3.27) and (3.3.28), let us take the two cases separately (e.g. $\Omega_S = 1$ and $\Omega_S = 0$):

- Case when $\Omega_S = 0$ (i.e. layout is parallel to y-axis):

Constraint (3.3.6) will be replaced with equations (3.3.33) and (3.3.34):

$$y_{c1} - y_{c2} \geq \frac{d_{c1} + d_{c2}}{2} \quad \forall\, c1, c2 \in C \tag{3.3.33}$$

$$y_{c2} - y_{c1} \geq \frac{d_{c1} + d_{c2}}{2} \quad \forall\, c1, c2 \in C \tag{3.3.34}$$

Constraints (3.3.27) and (3.3.28) will be replaced with equations (3.3.35) to (3.3.40):

$$Xp_{ij} \geq (x_i - x_c) + \left(x_c - x_j\right) \;\; \forall\, i, j, c \tag{3.3.35}$$

$$Xp_{ij} \geq (x_i - x_c) + (x_j - x_c) \ \forall\, i, j, c \tag{3.3.36}$$

$$Xp_{ij} \geq (x_c - x_i) + (x_c - x_j) \ \forall\, i, j, c \tag{3.3.37}$$

$$Xp_{ij} \geq (x_c - x_i) + (x_j - x_c) \ \forall\, i, j, c \tag{3.3.38}$$

$$Yp_{ij} \geq (y_i - y_j) \ \forall\, i, j \neq i \tag{3.3.39}$$

$$Yp_{ij} \geq (y_j - y_i) \ \forall\, i, j \neq i \tag{3.3.40}$$

- Case when $\Omega_S = 1$ (i.e. layout is parallel to x-axis):

Constraint (3.3.6) will be replaced with equations (3.3.41) and (3.3.42):

$$x_{c1} - x_{c2} \geq \frac{l_{c1} + l_{c2}}{2} \ \forall\, c1, c2 \in C \tag{3.3.41}$$

$$x_{c2} - x_{c1} \geq \frac{l_{c1} + l_{c2}}{2} \ \forall\, c1, c2 \in C \tag{3.3.42}$$

Constraints (3.3.27) and (3.3.28) will be replaced with equations (3.3.35) to (3.3.40):

$$Yp_{ij} \geq (y_i - y_c) + (y_c - y_j) \ \forall\, i, j \neq i, c \tag{3.3.43}$$

$$Yp_{ij} \geq (y_i - y_c) + (y_j - y_c) \ \forall\, i, j \neq i, c \tag{3.3.44}$$

$$Yp_{ij} \geq (y_c - y_i) + (y_c - y_j) \ \forall\, i, j \neq i, c \tag{3.3.45}$$

$$Yp_{ij} \geq (y_c - y_i) + (y_j - y_c) \ \forall\, i, j \neq i, c \tag{3.3.46}$$

$$Xp_{ij} \geq (x_i - x_j) \ \forall\, i, j \neq i \tag{3.3.47}$$

$$Xp_{ij} \geq (x_j - x_i) \ \forall\, i, j \neq i \tag{3.3.48}$$

Objective functions

The objective function is the same as in the previous formulation of section 3.2, where the aim is to minimize the total traveling cost and to maximize the desired closeness-rating factor based on international standards:

$$\boldsymbol{Min\ F} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{4} \boldsymbol{F_{ijk}(Xp_{ij} + Yp_{ij})\varphi_{ijk} * \sigma_k} - \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{R_{ij}\mu_{ij}} \tag{3.3.49}$$

3.3.2. Computational Experience

The proposed formulations are validated using a set of generated data instances. ILOG CPLEX 12.5 software is used to solve the data instances using Windows 7 platform, Intel5® Core ™ i5-3380M CPU@ 2.90GHz and 8Go of RAM.

Data Instances

Data instances were generated after observing the real operations at an OTL at a hospital, where statistics on the number of operations per day and per specialty were collected. The traveling data and other movements concerning an operation involving 1 patient, 1 doctor, 2 medical-staff and 1 non-medical staff) were estimated. For each instance, four sets of data were generated by modifying some parameters to test the robustness of the proposed models. Table 3.10 provides the notations of the data instances using a prefix indicated by SEC followed by the number of facilities (rooms), for each instance the associated types of rooms is listed.

Instance's size	Room list
SEC11	4 operating rooms, 1 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room and 2 corridors
SEC13	4 operating rooms, 2 induction rooms, 1 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room and 2 corridors
SEC16	4 operating rooms, 2 induction rooms, 2 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room, 1 changing room, 1 rest room and 2 corridors
SEC17	4 operating rooms, 2 induction rooms, 2 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room, 2 changing room, 1 bed storage and 2 corridors
SEC18	4 operating rooms, 2 induction rooms, 2 scrub room, 1 PACU, 1 decontamination, 2 sterile arsenal, 1 cleaning room, 2 changing room, 1 bed storage and 2 corridors
SEC20	6 operating rooms, 3 induction rooms, 2 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room, 2 changing room, 1 bed storage and 2 corridors

Table 3.10. Description of each multi-section instances

Several scenarios were considered in the generation of these instances. First, separating sections and affecting facilities to appropriate section according to international standards:

- Outer section
  - Areas for receiving patients, toilets, administrative function.
- Restricted section
  - Changing room
  - Patient transfer area
  - Stores room
  - Nursing staff room
  - Anesthetist room
  - Recovery room
  - PACU

- Aseptic section
  - Scrub area
  - Preparation room
  - Operation theatre
  - Area used for instrument packing and sterilization.
- Disposal section
  - Area where used equipment is cleaned and bio-hazardous waste is disposed.

Second, the size of the hospital is an important aspect for the dimension of its OT. For instance, in small OT (i.e. SEC11) the induction and the surgery are built inside the OR to reduce the number of rooms. For bigger OT, the number of induction rooms can vary from one design to another to serve all OR to one induction for each OR. The size of PACU also depends on the number of ORs. In fact, the PACU needs 1.5 to 2 beds for each OR while it does not exceed a limit of 10 beds, in this case, the separation of the PACU into two units is recommended (see [108] Anesthetic Safety Report).

Third, the exact method for solving large-sized layout limited us to the consideration of small OT. Therefore, we only considered the restricted and aseptic sections and the most important facilities that constitute each section, while the trip is assured by corridors to make possible for travel between these sections.

Finally, having a list of facilities is not sufficient to design the OT layout. It is necessary to determine whether or not a group of facilities could work together and with which criteria. Thus, three organization scenarios were adopted to highlight the constraints of space:

- Avoid designing a OT all in length, in order to reduce personal's traveling distances and their dispersion which acts on the collective functioning;
- Search for an equidistance between ORs and Stores room.
- Gather in homogeneous groups those facilities associated with the surgery on the first hand, and those related to recovery in the second hand; and place them according to the logic of the patient pathway.

In table 3.10, the structure SEC11 assumes that all operating rooms contain inside an induction service and that all rooms share the same scrub room. PACU, cleaning room, decontamination and arsenal are sharing the same section (restricted) according to aseptic technique. While in SEC13, large facilities and more relationships are considered. Here the induction rooms are considered independent, facilities where the OR1 and OR2 share the induction room1 while the OR3 and OR4 share the induction room2. The scrub room is also shared by all the facilities in the aseptic section.

For testing, we increased the size of instances, each time by one or two facilities until the optimal was not feasible to obtain. Our MILP formulation provides an optimal layout solution for the generated instances of size up to eighteen facilities. For instances up to twenty facilities, we used a fixed processing time for all sets. It can be observed that the solutions start to be away from optimality.

#	Data	Objective	Variables	Constraint	Time (sec)	Number of Iteration	%Gap
SEC11	Set 1	124427	657	1264	4	108343	0%
	Set 2	124523			7	129563	0%
	Set 3	125867			3	45028	0%
	Set 4	126855			2	40017	0%
SEC13	Set 1	233672	955	1872	25	509640	0%
	Set 2	232641			23	487562	0%
	Set 3	236347			8	199563	0%
	Set 4	236215			14	279385	0%
SEC16	Set 1	3251673	1507	3009	422	14510003	0%
	Set 2	3253471			612	15653586	0%
	Set 3	3265723			582	14520666	0%
	Set 4	3263600			877	17556056	0%
SEC17	Set 1	7535150	1719	3448	2612	52483377	0%
	Set 2	7535230			2889	52523486	0%
	Set 3	7569650			7967	91055295	0%
	Set 4	7566630			3225	60423216	0%
SEC18	Set 1	7576718	1945	3917	19638	263999558	0%
	Set 2	7633257			21647	293009506	0%
	Set 3	7635890			18996	217349500	0%
	Set 4	7515896			19124	215346500	0%
SEC20	Set 1	12936549	2439	4945	18000	369494150	21.51%
	Set 2	12878858			18000	366089050	21.16%
	Set 3	12585662			18000	448799249	16.45%
	Set 4	12562489			18000	515116160	15.59%

Table 3.11. Results of solving static multi-section OTLP

Table 3.11 shows the computational results for the continuous multi-section OTPL. The first column indicates the name of the test instance and the second shows set of data instances. The third column gives the costs for the generated layout plan. The number of variables and constraints are given in columns four and five respectively. Columns 6 and 7 show the time processing and the number of iterations to reach a final solution and the last one refers to percentage gap between the lower bound and the upper bound for each instance. The lower bound is obtained from the linear relaxation of the integer variables and the upper bound is the optimal value or the best solution obtained when the CPU time limit is reached.

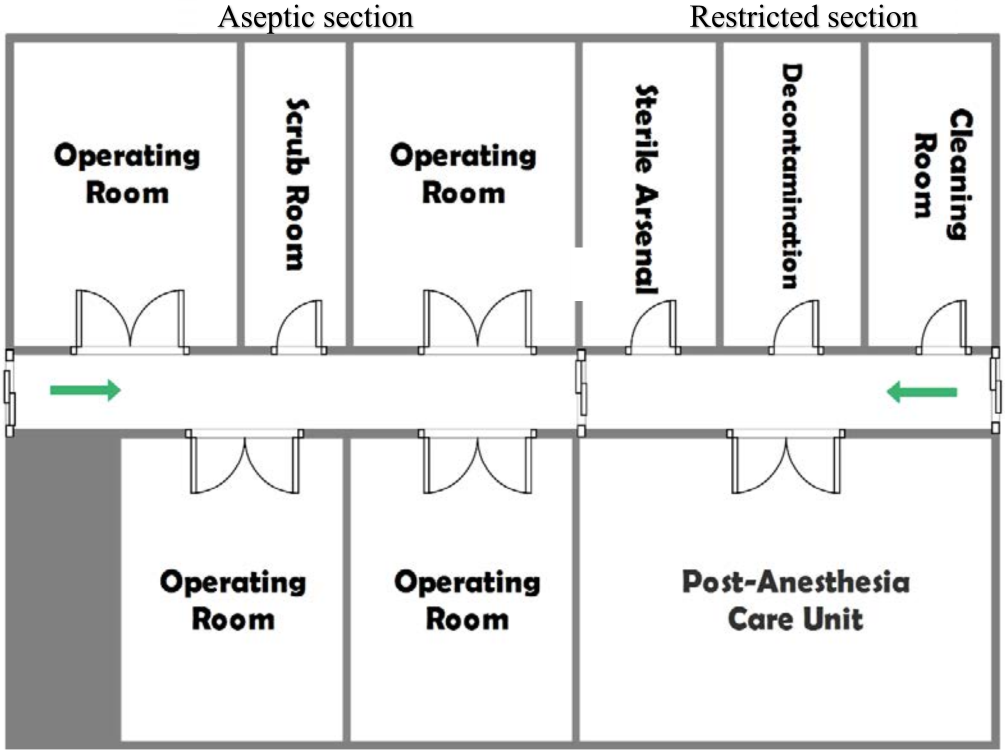


Figure 3.9. SEC11 optimal layout

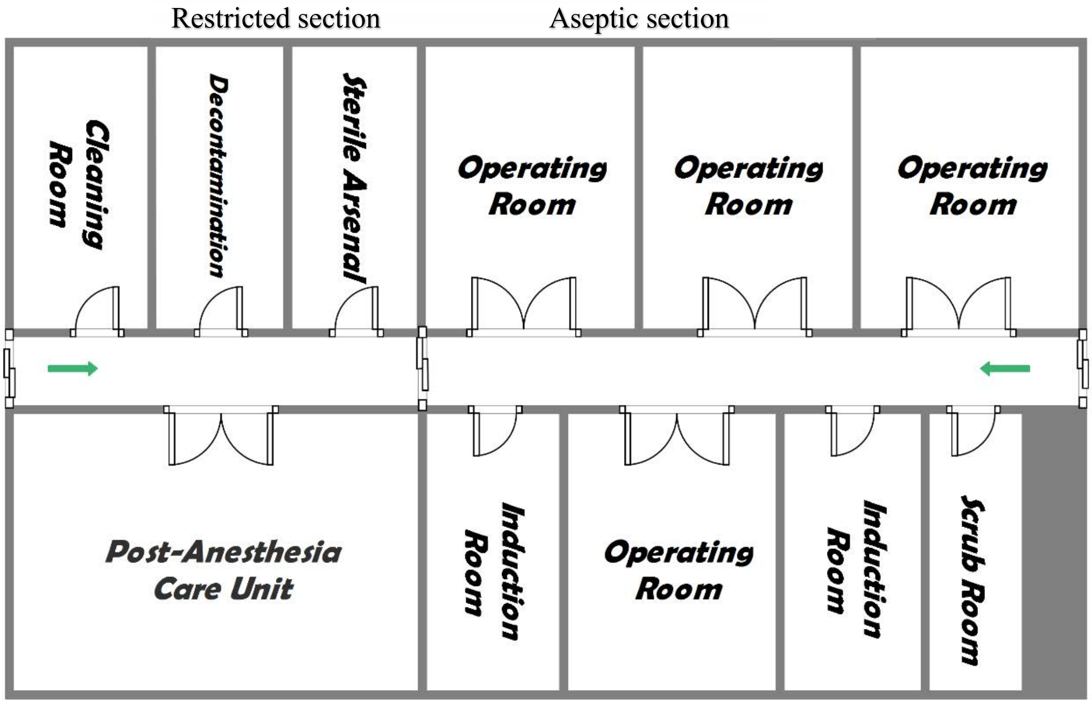


Figure 3.10. SEC13 optimal layout

The best solution of some problems is depicted in the following figures: Figure 3.9 represents the optimal OTL with 11 facilities (i.e. SEC11), Figure 3.10 gives the optimal OTL for SEC13 with 13 facilities while the solution for the SEC16 with 16 facilities is presented in Figure 3.11.

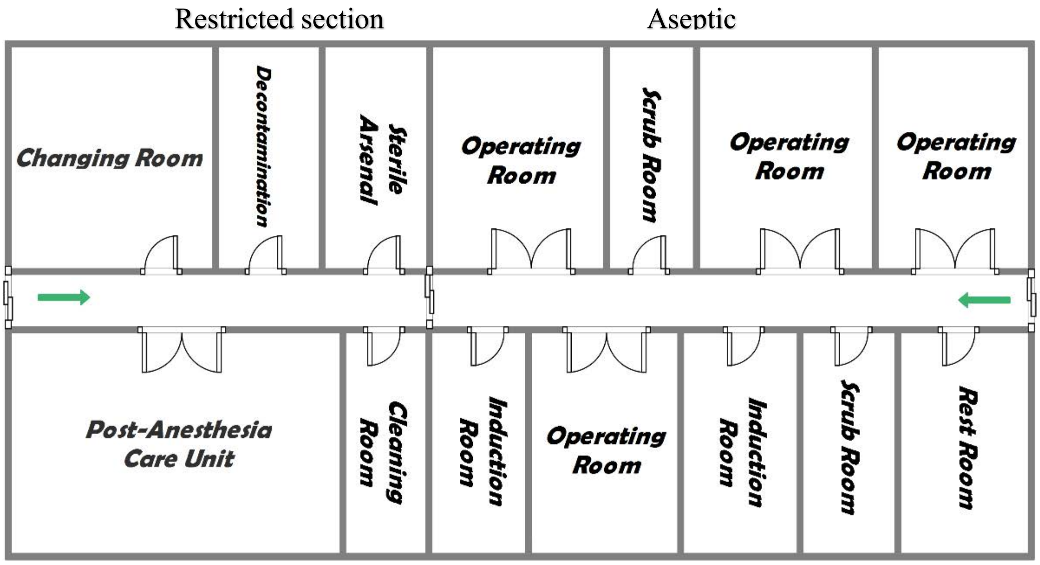


Figure 3.11. SEC16 optimal layout

3.3.3. Summary

This part describes a Mixed Integer Programing model for the Multi-Section OTLP and motivation of this formulation. The main objective is to minimize the total traveling cost and maximize the total closeness rating while respecting some restrictions according to international norms of OT's security and hygiene. This formulation is an extension of the previous one in Section (3.2). It was shown that the capacity of the exact method is capable to solve instances of 18 facilities. The formulation permitted us to address more realistic grouping criteria that regulate the OTs' design. The developed models can be used to build an automated decision support tool to aid planners to obtain their OT layout design. Our modeling approach would have a great impact on improving the design of future OT. It integrates professional standards, which was never considered in the past.

3.4. Mixed Integer Programming model for the Multi-Floor OTLP

In this section, another extension is considered. It addresses another type of structure in which the OT is divided across several floors due to limited availability of space, namely, **the mutli-floor OTLP**. In literature, this kind of problem is known as multi-floor FLP in industrial settings for which several works were addressed (cf. Chapter II).

In this section, we introduce the multi-floor OTLP within the health care settings. It will be formulate as a MIP with the objective to determine the locations of the facilities and elevators within each floor as to minimize the total traveling cost and to maximize the closeness rating.

3.4.1. Mathematical Formulation

Assumptions

In this formulation, we assume that:

- The number of floor, area of each floor and the number of elevators are all modelled as parameters;
- The elevators are modelled as facilities and consume layout's area;
- The elevators' location is a decision variable;
- Elevators have the same x and y coordinates on any floor;
- A facility cannot be divided among multiple floors;
- Vertical travel between floors can only occur through elevators;
- Elevator capacity is not considered;
- Corridors are used to travel between two facilities in the same floor, and elevators are used to transit from one to the other floor;
- Floor have the same shape, the same dimension and the same orientation;
- Floors are oriented parallel to x-axis.
- Each floor contains facilities with the same operating mode (i.e. floor for all surgery facilities, a second one with all recovery facilities, etc.)

Sets

- Let N= { a_i ; i=1,2,…,n } be the set of n facilities;
- Let K= { e_k ; k=1, …,4} be the set of k entity types;
- Let F= {ε_f ; s=1, 2,…,t } be the set of floors;
- Let C= {θ_c; c=1, 2,…,r } be the set of corridors;
- Let E= {ϑ_e; e=1, 2,…,o} be the set of elevators location;
- Let U_i be a set of a single element denoting the floor to which facility a_i is belonging.
- Let U_c be a set of single element denoting the floor to which corridor θ_c is belonging.
- Let U_f be a set of single element denoting the floor to which elevator ε_f is belonging.

Parameters

α_i :	Length of facility a_i
β_i :	Width of facility a_i
l_c, d_c:	Dimensions of corridor θ_c
l_e, d_e:	Dimensions of elevator ϑ_e
H	Common floor height
F_{ijk}:	Number of trips between facility a_i to facility a_j made by an entity type e_k

φ_{ijk}: Moving difficulty between facility a_i to facility a_j made by an entity type e_k

σ_k: Cost of horizontal traveling by entity e_k

σ_e: Cost of vertical traveling when elevator ϑ_e is used

x_{max}: Maximum length of floors ;

y_{max}: Maximum width of floors ;

R_{ij}: Desirable relationship value between facility a_i to facility a_j;

ρ_1, ρ_2: Weights for each sub-objective function.

Decision Variables

Ω_i { 1 if length α_i of facility a_i is parallel to x-axis
0 otherwise

μ_{ij} { 1 if a_i and a_j are fully adjacent
0 otherwise

V_{ije} { 1 if traffic between facilities a_i and a_j travels through elevator ϑ_e
0 otherwise

t_{ij} { 1 if facility a_i and a_j are assigned to the same floor
0 otherwise

t_{ic} { 1 if facility a_i and corridor θ_c are assigned to the same floor
0 otherwise

v_{if} { 1 if facility a_i is assigned to floor ε_f
0 otherwise

v_{cf} { 1 if corridor θ_c is assigned to floor ε_f
0 otherwise

v_{ef} { 1 if elevator ϑ_e is assigned to floor ε_f
0 otherwise

z_{ij}^x { 1 if facility a_i is strictly to the right of facility a_j
0 otherwise

z_{ij}^y { 1 if facility a_i is strictly above facility a_j
0 otherwise

o_{ic}^x { 1 if facility a_i is strictly to the right of corridor θ_c
0 otherwise

o_{ic}^{y}	{	1 if facility a_i is strictly above corridor θ_c 0 otherwise
ω_{if}^{x}	{	1 if facility a_i is strictly to the right of elevator ϑ_e 0 otherwise
ω_{cf}^{y}	{	1 if elevator ϑ_e is strictly above corridor θ_c 0 otherwise
x_i , y_i:		x and y coordinates of the geometric center of gravity facility a_i
x_c ,y_c :		coordinates of the geometrical center of corridor θ_c
x_e ,y_e :		coordinates of the geometrical center of elevator ϑ_e
l_i:		x-length of facility a_i depending on whether α_i or β_i is parallel on x-axis
d_i:		y-length of facility a_i depending on whether α_i or β_i is parallel on y-axis
Xp_{ij}:		x-distance between facility a_i and a_j in the same floor
Yp_{ij}:		y-distance between facility a_i and a_j in the same floor
Xe_{ie}:		x-distance between facility a_i and elevator ϑ_e
Ye_{ie}:		y-distance between facility a_i and elevator ϑ_e
Ze_{ij}:		Vertical distance between facility a_i and a_j in different floors

Constraints

a) Orientation constraints

These constraints are the same in (3.2.1) and (3.2.2):

$$l_i = \alpha_i \Omega_i + \beta_i (1 - \Omega_i) \qquad \forall i \tag{3.4.1}$$

$$d_i = \alpha_i + \beta_i - l_i \qquad \forall i \tag{3.4.2}$$

b) Floors constraints

Constraints (3.4.3) ensure that each facility is assigned to only one floor. Constraint (3.4.4) is used to obtain the value of t_{ij}: If two facilities a_i and a_j are allocated to the same floor ($v_{if} = v_{jf} = 1$), then constraint (3.4.4) will assure that t_{ij}=1. Else, if facilities a_i and a_j are allocated to different floors, then constraint will ensure that t_{ij}=0. This constraint is linearized using equations (3.4.5) to (3.4.7):

$$\sum_{f=1}^{F} v_{if} = 1 \qquad \forall i \tag{3.4.3}$$

$$t_{ij} = \left|v_{if} + v_{jf} - 1\right| \;\; \forall\, i, j \neq i, f \tag{3.4.4}$$

$$t_{ij} \geq v_{if} + v_{jf} - 1 \;\; \forall\, i, j \neq i, f \tag{3.4.5}$$

$$t_{ij} \leq 1 - v_{if} + v_{jf} \;\; \forall\, i, j \neq i, f \tag{3.4.6}$$

$$t_{ij} \leq 1 + v_{if} - v_{jf} \;\; \forall\, i, j \neq i, f \tag{3.4.7}$$

We do the same for corridors, (3.4.8) ensure that each corridor is assigned to only one floor, while constraints (3.4.9)-(3.4.11) are used to obtain the value of t_{ij}: If a facility a_i and a corridor are allocated to the same floor ($v_{if} = v_{cf} = 1$), then constraint (3.4.9) will have $t_{ic}=1$, while constraints (3.4.10) and (3.4.11) remain inactive. Else, if facility a_i and corridor are allocated to different floors, then constraints (3.4.9), (3.4.10) and (3.4.11) will ensure that $t_{ic}=0$.

$$\sum_{f=1}^{F} v_{cf} = 1 \qquad \forall i \tag{3.4.8}$$

$$t_{ic} \geq v_{if} + v_{cf} - 1 \;\; \forall\, i, c, f \tag{3.4.9}$$

$$t_{ic} \leq 1 - v_{if} + v_{cf} \;\; \forall\, i, c, f \tag{3.4.10}$$

$$t_{ic} \leq 1 + v_{if} - v_{cf} \;\; \forall\, i, c, f \tag{3.4.11}$$

c) Elevators constraints

Constraint (3.4.12) ensures that the routing of the flow between two facilities is through elevator in the case when they are not assigned to the same floor.

$$\sum_{e=1}^{E} V_{ije} = 1 - t_{ij} \;\; \forall\, i, j \neq i \tag{3.4.12}$$

The elevator location in each floor must have vertically. Thus, constraint (3.4.13) assures that elevators have the same x coordinate in each floor, while constraint (3.4.14) forces elevators to have the same y coordinate in each floor. Constraint (3.4.15) makes sure that each elevator is assigned to only one floor.

$$x_{e1} = x_{e2} \qquad \forall\, e1 = 1, \dots, E-1;\;\; e2 = e1+1, \dots, E \tag{3.4.13}$$

$$y_{e1} = y_{e2} \qquad \forall\, e1 = 1, \dots, E-1;\;\; e2 = e1+1, \dots, E \tag{3.4.14}$$

$$\sum_{e=1}^{E} v_{ef} = 1 \qquad \forall f \tag{3.4.15}$$

d) Facilities non-overlapping constraints

As in the previous formulation, we use the mixed integer constraints (3.4.16) to (3.4.20) to insure that two facilities must either not overlap on the same floor or be located on different floors:

$$x_i - x_j + X_{max}\left(2 - z_{ij}^x - t_{ij}\right) \geq \frac{l_i + l_j}{2} \quad \forall\, i, j \neq i \tag{3.4.16}$$

$$y_i - y_j + Y_{max}\left(2 - z_{ij}^y - t_{ij}\right) \geq \frac{d_i + d_j}{2} \quad \forall\, i, j \neq i \tag{3.4.17}$$

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \geq t_{ij} \;\; \forall\, i, j \neq i \tag{3.4.18}$$

$$z_{ij}^x + z_{ji}^x \leq t_{ij} \;\; \forall\, i, j \neq i \tag{3.4.19}$$

$$z_{ij}^y + z_{ji}^y \leq t_{ij} \;\; \forall\, i, j \neq i \tag{3.4.20}$$

e) Bounding constraints

Constraints (3.4.21) through (3.4.24) indicate that facilities have to be allocated within the appropriate floor space defined by the corners (0, 0) and (X_{max}, Y_{max}), equations (3.4.25) through (3.4.28) insure the same bounding for corridors and equations (3.4.29) to (3.4.32) force elevators to be located within floor space.

$$x_i + \frac{l_i}{2} \leq X_{max} \quad \forall\, i \tag{3.4.21}$$

$$y_i + \frac{d_i}{2} \leq Y_{max} \quad \forall\, i \tag{3.4.22}$$

$$x_i - \frac{l_i}{2} \geq 0 \quad \forall\, i \tag{3.4.23}$$

$$y_i - \frac{d_i}{2} \geq 0 \quad \forall\, i \tag{3.4.24}$$

$$x_c + \frac{l_c}{2} \leq X_{max} \quad \forall\, c \tag{3.4.25}$$

$$y_c + \frac{d_c}{2} \leq Y_{max} \quad \forall\, c \tag{3.4.26}$$

$$x_c - \frac{l_c}{2} \geq 0 \quad \forall\, c \tag{3.4.27}$$

$$y_c - \frac{d_c}{2} \geq 0 \quad \forall\, c \tag{3.4.28}$$

$$x_e + \frac{l_e}{2} \leq X_{max} \quad \forall\, e \tag{3.4.29}$$

$$y_e + \frac{d_e}{2} \leq Y_{max} \quad \forall\, e \tag{3.4.30}$$

$$x_e - \frac{l_e}{2} \geq 0 \quad \forall\, e \tag{3.4.31}$$

$$y_e - \frac{d_e}{2} \geq 0 \quad \forall\, e \tag{3.4.32}$$

f) Corridors & facilities non-overlapping constraints

Constraints (3.4.33) to (3.4.37) provide the non-overlapping between a facility a_i and a corridor θ_c if they occupy the same floor; or to be in different floors.

$$\left(x_c + \frac{l_c}{2}\right) \leq x_i + \frac{l_i}{2} + X_{max}(2 - o_{ic}^x - t_{ic}) \ \forall\, i, c \tag{3.4.33}$$

$$\left(x_c - \frac{l_c}{2}\right) + X_{max}(2 - o_{ci}^x - t_{ic}) \geq x_i + \frac{l_i}{2} \ \forall\, i, c \tag{3.4.34}$$

$$\left(y_c + \frac{d_c}{2}\right) \leq y_i + \frac{d_i}{2} + Y_{max}\left(2 - o_{ic}^y - t_{ic}\right) \ \forall\, i, c \tag{3.4.35}$$

$$\left(y_c - \frac{d_c}{2}\right) + Y_{max}\left(2 - o_{ci}^y - t_{ic}\right) \geq y_i + \frac{d_i}{2} \ \forall\, i, c \tag{3.4.36}$$

$$o_{ic}^x + o_{ci}^x + o_{ic}^y + o_{ci}^y \geq t_{ic} \quad \forall\, i, c \tag{3.4.37}$$

g) Facilities and elevators non-overlapping constraints

Here, we use a mixed integer constraints (3.4.38) to (3.4.42) to insure that a facility and an elevator must not overlap on the same floor:

$$x_i - x_e + X_{max}(1 - \omega_{ie}^x) \geq \frac{l_i + l_e}{2} \quad \forall\, i, \forall\, e \tag{3.4.38}$$

$$y_i - y_e + Y_{max}\left(1 - \omega_{ie}^y\right) \geq \frac{d_i + d_e}{2} \quad \forall\, i, \forall\, e \tag{3.4.39}$$

$$\omega_{ie}^x + \omega_{ei}^x + \omega_{ie}^y + \omega_{ei}^y \geq 1 \quad \forall\, i, \forall\, e \tag{3.4.40}$$

$$\omega_{ie}^{x} + \omega_{ei}^{x} \leq 1 \quad \forall\, i, \forall\, e \tag{3.4.41}$$

$$\omega_{ie}^{y} + \omega_{ei}^{y} \leq 1 \quad \forall\, i, \forall\, e \tag{3.4.42}$$

h) Distance constraints

For distance, we calculate the travel between any two facilities by going through corridors using Manhattan norm with constraints (3.4.43) and (3.4.48) if the two facilities are located in the same floor:

$$Yp_{ij} \geq (y_i - y_c) + (y_c - y_j) - (1 - t_{ic})(X_{max} + Y_{max})\ \forall\, i, j, c \tag{3.4.43}$$

$$Yp_{ij} \geq (y_i - y_c) + (y_j - y_c) - (1 - t_{ic})(X_{max} + Y_{max})\ \forall\, i, j, c \tag{3.4.44}$$

$$Yp_{ij} \geq (y_c - y_i) + (y_c - y_j) - (1 - t_{ic})(X_{max} + Y_{max})\ \forall\, i, j, c \tag{3.4.45}$$

$$Yp_{ij} \geq (y_c - y_i) + (y_j - y_c) - (1 - t_{ic})(X_{max} + Y_{max})\ \forall\, i, j, c \tag{3.4.46}$$

$$Xp_{ij} \geq (x_i - x_j) - (1 - t_{ij})(X_{max} + Y_{max}) \quad \forall\, i, j \neq i \tag{3.4.47}$$

$$Xp_{ij} \geq (x_j - x_i) - (1 - t_{ij})(X_{max} + Y_{max})\ \forall\, i, j \neq i \tag{3.4.48}$$

The distance between a facility and each elevator is obtained from constraints (3.4.49) to (3.4.54):

$$Ye_{ie} \geq (y_i - y_c) + (y_c - y_e) \ \ \forall\, i, e, c \tag{3.4.49}$$

$$Ye_{ie} \geq (y_i - y_c) + (y_e - y_c) \ \ \forall\, i, e, c \tag{3.4.50}$$

$$Ye_{ie} \geq (y_c - y_i) + (y_c - y_e)\ \forall\, i, e, c \tag{3.4.51}$$

$$Ye_{ie} \geq (y_c - y_i) + (y_e - y_c) \ \ \forall\, i, e, c \tag{3.4.52}$$

$$Xe_{ie} \geq (x_i - x_e) \ \ \forall\, i, e \tag{3.4.53}$$

$$Xe_{ie} \geq (x_e - x_i)\ \forall\, i, e \tag{3.4.54}$$

Constraints (3.4.55) and (3.4.56) calculate the vertical distances between two facilities according to the height of each floor and the number of floors separating the two facilities. When the facilities are located on the same floor (i.e. $v_{if} = v_{jf}$), the vertical distance is equal to zero.

$$Ze_{ij} \geq H \sum_{b=1}^{F} b\left(v_{if} - v_{jf}\right) \quad \forall\, i, j \neq i, f \tag{3.4.55}$$

$$Ze_{ij} \geq H \sum_{b=1}^{F} b\left(v_{jf} - v_{if}\right) \;\; \forall\, i, j \neq i, f \tag{3.4.56}$$

Constraint (3.4.57) and (3.4.58) compute the total distance between the two departments as the sum of the distance between each facility and the elevator determined by constraints (3.4.49)-(3.4.54). This constraint is only active when facilities a_i and a_j are assigned to different floors (t_{ij} =0) and when elevator is used to travel between the two facilities (V_{ije} = 1):

$$Yp_{ij} \geq Ye_{ie} + Ye_{je} - [t_{ij}Y_{max} + \left(1 - V_{ije}\right)Y_{max}] \quad \forall\, i, j, e \tag{3.4.57}$$

$$Xp_{ij} \geq Xe_{ie} + Xe_{je} - [t_{ij}X_{max} + \left(1 - V_{ije}\right)X_{max}] \quad \forall\, i, j, e \tag{3.4.58}$$

Objective functions

The objective function is the same in previous formulation, where we aim to minimize the total traveling cost and to maximize the desired closeness-rating factor based on international standards:

$$\boldsymbol{min\, F} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{4} \boldsymbol{F_{ijk}}\left[\left(\boldsymbol{Xp_{ij}} + \boldsymbol{Yp_{ij}}\right)\boldsymbol{\varphi_{ijk}} * \boldsymbol{\sigma_k} + \boldsymbol{Ze_{ij}\sigma_e}\right] - \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{R_{ij}\mu_{ij}} \tag{3.4.59}$$

3.4.2. Experiments and results

The proposed formulations are validated using a set of generated data on OTFL optimization. We used ILOG CPLEX 12.5 software to solve the model using Windows 7 platform, Intel5® Core ™ i5-3380M CPU@ 2.90GHz and 8Go of RAM.

The use of exact method limited us to integrate a finite number of facilities to deal with small-sized OTLP. In addition to that, corridors and elevators are considered as facilities where location is a decision variable. Thus, the choice of facilities to consider in our formulation is based on real observation and professional recommendation to select the necessary facilities for each instance's size. Table 3.12 provides the notations of the test instances prefixed by FLR and gives for each instance the list of associated type of rooms.

As in the previous formulation, several considerations and restrictions are taking into account to group facilities according to their operating properties:

- For experiments, we use two floors, each one groups facilities with link to a function of the OT.
- Facilities in first floor are concerned with surgery and pre-surgery phases, while the second one contains post-surgery phase and administration facilities.
- From observation, the surgery facilities are often located in the underground floor.

Table 3.13 shows the computational results for the continuous mutli-floor OTPL. The best solution of some problems is dressed in the following figures: Figure 3.12 represents the optimal OTL with 14 facilities (i.e. FLR14); Figure 3.13 gives the optimal OTL for FLR16 with 16 facilities while FLR18 with 18 facilities is designed in Figure 3.14.

Instance's size	Room list
FLR14	4 operating rooms, 1 induction rooms, 1 scrub room, 1 PACU, 1 sterile arsenal , 1 monitoring, 1 recovery rooms, 2 corridors and 2 elevators
FLR16	4 operating rooms, 1 induction rooms, 1 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal , 1 monitoring, 2 recovery rooms, 2 corridors and 2 elevators
FLR18	4 operating rooms, 1 induction rooms, 1 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room, 1 bed storage, 1 monitoring, 2 recovery rooms, 2 corridors and 2 elevators
FLR20	4 operating rooms, 2 induction rooms, 2 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room, 1 bed storage, 1 monitoring, 2 recovery rooms, 2 corridors and 2 elevators
FLR22	4 operating rooms, 2 induction rooms, 2 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room, 2 changing room, 1 bed storage, 1 monitoring, 2 recovery rooms, 2 corridors and 2 elevators

Table 3.12. Description of each multi-floor instances

3.1.1. Summary

This part describes a Mixed Integer Programing model for the Multi-Floor OTLP and motivation of this formulation. The main objective is to minimize the total traveling cost and maximize the total closeness rating while respecting some restrictions and international norms of OT's security and hygiene. This formulation is an extension of the previous one by addressing another case of realistic grouping criteria that regulate the OTs' design. The next part will describe an extension of this formulation to address another type of structure in which the OT is divided across several floors: ***the mutli-row OTLP****.*

#	Data	Objective	Variables	Constraint	Time (sec)	Number of Iteration	%Gap
FLR14	Set 1	5531700	3355	5785	2729	52483377	0%
	Set 2	5531583			2759	52596475	0%
	Set 3	5175250			5048	89055295	0%
	Set 4	5156254			4940	78473936	0%
FLR16	Set 1	7957500	4169	8076	10403	105496574	0%
	Set 2	8046800			12412	153009506	0%
	Set 3	6856500			9905	147349500	0%
	Set 4	6547300			9943	130248300	0%
FLR18	Set 1	7957500	4891	9226	10212	185476574	0%
	Set 2	8145200			12536	209508702	0%
	Set 3	7852300			9897	178630500	0%
	Set 4	7245800			9858	175045200	0%
FLR20	Set 1	9557100	5601	9853	18000	263889458	5.06%
	Set 2	9635400			18000	235019126	8.56%
	Set 3	9437892			18000	285869500	3.65%
	Set 4	9556254			18000	241206200	5.71%
FLR22	Set 1	10565800	6458	10580	18000	363819126	9.23%
	Set 2	11548625			18000	330569500	10.55%
	Set 3	12036589			18000	323389458	10.96%
	Set 4	10256932			18000	352508230	8.71%

Table 3.13. Results of solving static multi-floor OTLP

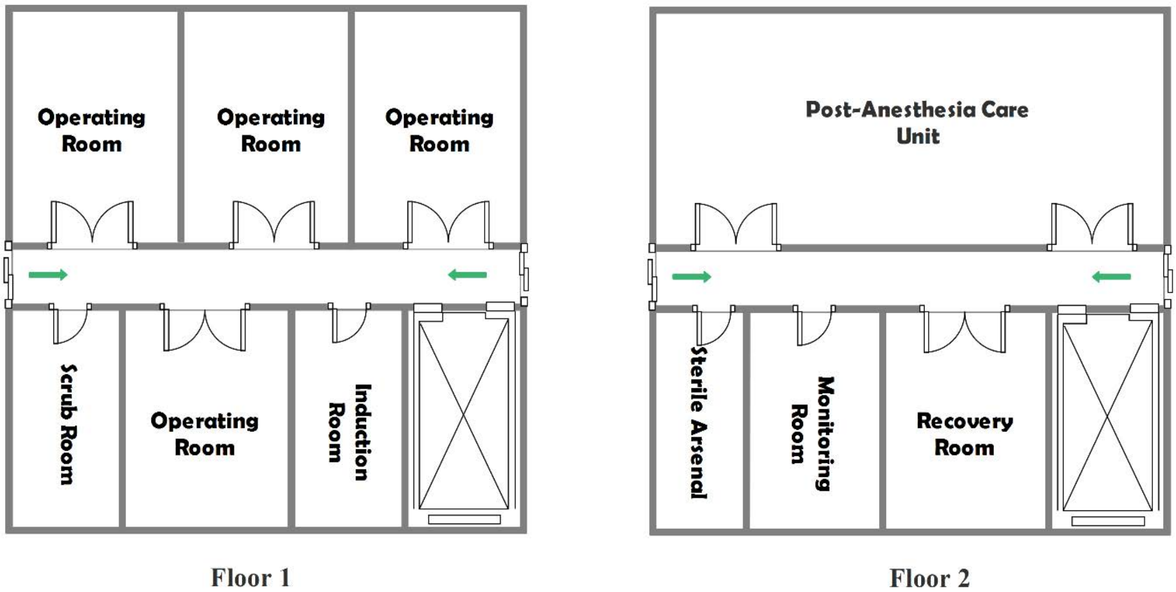
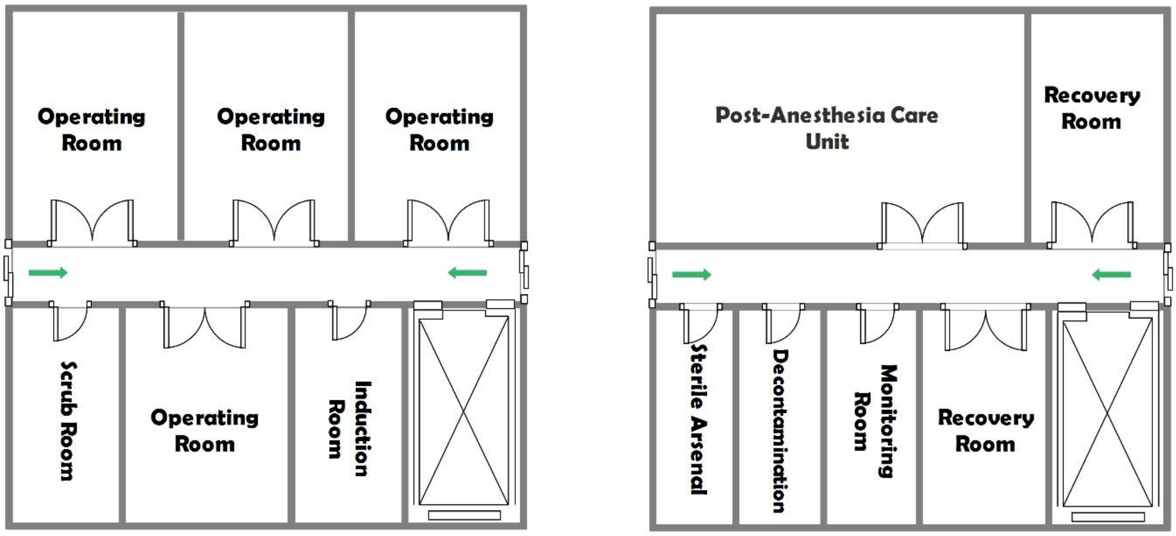
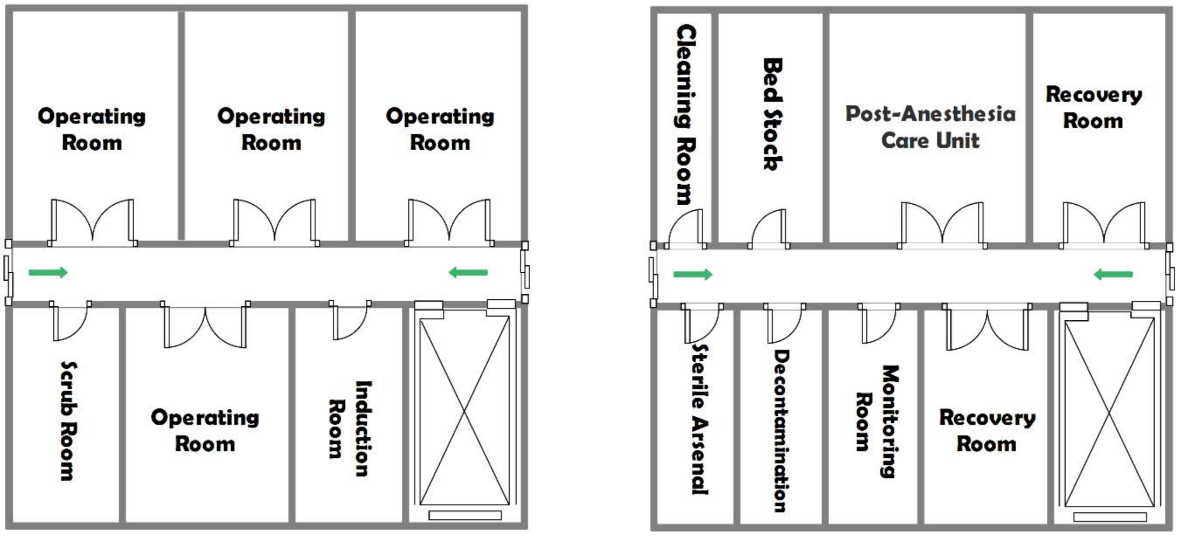


Figure 3.12. FLR14 optimal layout



Floor 1 Floor 2

Figure 3.13. FLR16 optimal layout



Floor 1 Floor 2

Figure 3.14. FLR18 optimal layout

3.5. Mixed Integer Programming model for the Multi-Row OTLP

In the previous formulations, corridors location was a decision variable, which makes the problem more complex by adding more constraints and more solution space to browse. In this formulation, we propose to fix the corridors' location and consider the problem as a multi-row problem (see Figure 3.15) for which several works in industrial context were addressed (cf. Chapter II).

In this part, we introduce the multi-row OTLP and formulate it as a MIP with the objective to determine the locations of the facilities within each row as to minimize the total traveling cost and to maximize the closeness rating.

Row 1	8	7	2	3	1
	Corridor				
Row 2	9	6	4	5	10
Row 3	18	17	12	13	11
	Corridor				
Row 4	19	16	14	15	11

Figure 3.15. Multi-row facility layout problem

3.5.1. Mathematical Formulation

Assumptions

For the multi-row OTLP, we assume:

- Each facility is located in row paralleling to horizontal axis;
- All facilities located in the same row have the same y-axis value;
- Corridors are located parallel to horizontal axis and their location is known;
- All rows have the same width and the same length;
- The number of row and row is known;
- Inter-facilities distance is computed using Manhattan norm while travel is ensured by corridors;
- A vertical corridor is considered to allow routing between different rows; we call it central corridor' and give it index c = 1.

Sets

- Let N= $\{a_i\,;\, i=1,2,\ldots,n\}$ be the set of n facilities;
- Let K= $\{e_k\,;\, k=1, \ldots,4\}$ be the set of k entity types;
- Let C= $\{\theta_c$; c=1, 2,…,r$\}$ be the set of corridors;
- Let R= $\{r_r$; e=1, 2,…,v$\}$ be the set of rows;
- Let U_i be a set of a single element denoting the row to which facility a_i is belonging.

Parameters

α_i : Length of facility a_i

β_i : Width of facility a_i

l_c, d_c: Dimensions of corridor θ_c

L_r, D_r: Dimensions of row r_r

F_{ijk}: Number of trips between facility a_i to a_j made by an entity type e_k

φ_{ijk}: Moving difficulty between facility a_i to a_j made by an entity type e_k

σ_k: Cost of horizontal traveling by entity e_k

x_{max}: Maximum length of OT ;

y_{max}: Maximum width of OT ;

x_c ,y_c : coordinates of the geometrical center of corridor θ_c

x_r ,y_r : coordinates of the geometrical center of row r_r

R_{ij}: Desirable relationship value between facility a_i to facility a_j;

ρ_1, ρ_2: Weights for each sub-objective function.

H_c Distance between two successive corridors

Decision Variables

Ω_i $\begin{cases} 1 \text{ if length } (\alpha_i) \text{ of facility } a_i \text{ is parallel to x-axis (horizontal orientation)} \\ 0 \text{ otherwise} \end{cases}$

μ_{ij} $\begin{cases} 1 \text{ if } a_i \text{ and } a_j \text{ are fully adjacent} \\ 0 \text{ otherwise} \end{cases}$

t_{ij} $\begin{cases} 1 \text{ if facility } a_i \text{ and } a_j \text{ are assigned to the same row} \\ 0 \text{ otherwise} \end{cases}$

v_{ir} $\begin{cases} 1 \text{ if facility } a_i \text{ is assigned to row } r_r \\ 0 \text{ otherwise} \end{cases}$

v_{ic} $\begin{cases} 1 \text{ if facility } a_i \text{ is adjacent to corridor } \theta_c \\ 0 \text{ otherwise} \end{cases}$

o_{ij} $\begin{cases} 1 \text{ if facilities } a_i \text{ and } aj \text{ are adjacent to the same corridor } \theta_c \\ 0 \text{ otherwise} \end{cases}$

z_{ij}^x $\begin{cases} 1 \text{ if facility } a_i \text{ is strictly to the right of facility } a_j \\ 0 \text{ otherwise} \end{cases}$

x_i , y_i: x and y coordinates of the geometric center of gravity facility a_i

l_i:	x-length of facility a_i depending on whether α_i or β_i is parallel on x-axis
d_i:	y-length of facility a_i depending on whether α_i or β_i is parallel on y-axis
Xp_{ij}:	x-distance between facility a_i and a_j in the same floor
Yp_{ij}:	y-distance between facility a_i and a_j in the same floor
Xc_i:	x-distance between facility a_i and central corridor θ_c
Y_{c1c2}:	Vertical distance between corridor θ_{c2} and θ_{c2}

Constraints

a) Orientation constraints

These constraints are the same in (3.2.1) and (3.2.2):

$$l_i = \alpha_i \Omega_i + \beta_i (1 - \Omega_i) \qquad \forall i \tag{3.5.1}$$

$$d_i = \alpha_i + \beta_i - l_i \qquad \forall i \tag{3.5.2}$$

b) Rows constraints

Constraints (3.5.3) ensure that each facility is assigned to only one row. Constraint (3.5.4) is used to obtain the value of t_{ij}: If two facilities a_i and a_j are allocated to the same row ($v_{ir} = v_{jr} = 1$), then constraint (3.5.4) will ensure that t_{ij}=1. Else, if facilities a_i and a_j are allocated to different rows, then the same constraint will ensure that t_{ij}=0. This constraint is linearized using equations (3.5.5) to (3.5.7):

$$\sum_{r=1}^{R} v_{ir} = 1 \qquad \forall i \tag{3.5.3}$$

$$t_{ij} = \left| v_{ir} + v_{jr} - 1 \right| \; \forall\, i, j \neq i, r \tag{3.5.4}$$

$$t_{ij} \geq v_{ir} + v_{jr} - 1 \;\; \forall\, i, j \neq i, r \tag{3.5.5}$$

$$t_{ij} \leq 1 - v_{ir} + v_{jr} \;\; \forall\, i, j \neq i, r \tag{3.5.6}$$

$$t_{ij} \leq 1 + v_{ir} - v_{jr} \;\; \forall\, i, j \neq i, r \tag{3.5.7}$$

c) Facilities and corridors adjacency

Constraints (3.5.8) ensure that each facility is adjacent to only one corridor. Constraint (3.5.9) to (3.5.11) are used to obtain the value of o_{ij} (i.e. if two facilities are adjacent to the same corridor or not): If two facilities a_i and a_j are adjacent to the same corridor θ_c ($v_{ic} = v_{jc} = 1$), then constraints will insure that o_{ij}=1. Else, if facilities a_i and a_j are adjacent to different corridors, then constraints will insure that o_{ij}=0.

$$\sum_{c=2}^{C} v_{ic} = 1 \qquad \forall i \tag{3.5.8}$$

$$o_{ij} \geq v_{ic} + v_{jc} - 1 \ \forall\, i, j \neq i, c \neq 1 \tag{3.5.9}$$

$$o_{ij} \leq 1 - v_{ic} + v_{jc} \ \forall\, i, j \neq i, c \neq 1 \tag{3.5.10}$$

$$o_{ij} \leq 1 + v_{ic} - v_{jc} \ \forall\, i, j \neq i, c \neq 1 \tag{3.5.11}$$

d) Facilities non-overlapping constraints

As in the previous formulation, we use the mixed integer constraints (3.5.12) to (3.5.14) to insure that two facilities must either not overlap on the same floor or be located on different floors:

$$x_i - x_j + X_{max}\left(2 - z_{ij}^{x} - t_{ij}\right) \geq \frac{l_i + l_j}{2} \quad \forall\, i, j \neq i \tag{3.5.12}$$

$$x_j - x_i + X_{max}\left(2 - z_{ij}^{x} - t_{ij}\right) \geq \frac{l_i + l_j}{2} \quad \forall\, i, j \neq i \tag{3.5.13}$$

$$z_{ij}^{x} + z_{ji}^{x} \leq t_{ij} \ \forall\, i, j \neq i \tag{3.5.14}$$

e) Bounding constraints

Constraints (3.5.15) through (3.5.18) indicate that facilities have to be allocated within the appropriate floor space defined by the corners (0, 0) and (L_r, D_r).

$$x_i + \frac{l_i}{2} \leq L_r \quad \forall\, i \tag{3.5.15}$$

$$y_i + \frac{d_i}{2} \leq D_r \quad \forall\, i \tag{3.5.16}$$

$$x_i - \frac{l_i}{2} \geq 0 \quad \forall\, i \tag{3.5.17}$$

$$y_i - \frac{d_i}{2} \geq 0 \quad \forall\, i \tag{3.5.18}$$

f) Distance constraints

For distance, we calculate the travel between any two facilities by going through corridors using Manhattan norm with constraints (3.5.19) and (3.5.19) if the two facilities are located in the same row or adjacent to the same corridor:

$$Yp_{ij} \geq (y_i - y_c) + \left(y_c - y_j\right) \forall\, i, j \neq i, c \neq 1 \tag{3.5.19}$$

$$Yp_{ij} \geq (y_i - y_c) + (y_j - y_c) \ \forall\, i, j \neq i, c \neq 1 \tag{3.5.20}$$

$$Yp_{ij} \geq (y_c - y_i) + (y_c - y_j) \ \forall\, i, j \neq i, c \neq 1 \tag{3.5.21}$$

$$Yp_{ij} \geq (y_c - y_i) + (y_j - y_c) \ \forall\, i, j \neq i, c \neq 1 \tag{3.5.22}$$

$$Xp_{ij} \geq (x_i - x_j) - (1 - t_{ij})(X_{max} + Y_{max}) \ \forall\, i, j \neq i \tag{3.5.23}$$

$$Xp_{ij} \geq (x_j - x_i) - (1 - t_{ij})(X_{max} + Y_{max}) \ \forall\, i, j \neq i \tag{3.5.24}$$

The distance between a facility and the central corridor is obtained from constraints (3.5.25) and (3.5.26):

$$Xc_i \geq (x_i - x_c) \ \forall\, i, c = 1 \tag{3.5.25}$$

$$Xc_i \geq (x_c - x_i) \ \forall\, i \text{, c= 1} \tag{3.5.26}$$

Constraints (3.5.27) and (3.5.28) calculate the vertical distances between two facilities according to the distance separating two successive corridors. When the facilities are adjacent to the same corridor (i.e. $v_{ic} = v_{jc}$), the use of central corridor is unneeded and vertical distance is equal to zero.

$$Y_{c1c2} \geq H_c \sum_{b=1}^{C} b(v_{ic} - v_{jc}) \ \forall\, i, j \neq i, c \neq 1 \tag{3.5.27}$$

$$Y_{c1c2} \geq H_c \sum_{b=1}^{C} b(v_{jc} - v_{ic}) \ \forall\, i, j \neq i, c \neq 1 \tag{3.5.28}$$

Constraint (3.5.29) compute the total distance between the two departments as the sum of the distance between each facility and the central corridor. This constraint is only active when facilities a_i and a_j are assigned to different row (t_{ij} =0) and when facilities are not adjacent to the same corridor (v_{ic} = 0):

$$Xp_{ij} \geq Xc_i + Xc_j - (t_{ij} + v_{ic})X_{max} \ \ \forall\, i, j \neq i, c \neq 1 \tag{3.5.29}$$

Objective functions

The objective function aims to minimize the total traveling cost and to maximize the desired closeness-rating factor based on international standards:

$$\boldsymbol{min\ F} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{4} \boldsymbol{F_{ijk}} \, (\boldsymbol{Xp_{ij}} + \boldsymbol{Yp_{ij}} + \boldsymbol{Y_{c1c2}})\boldsymbol{\varphi_{ijk}} * \boldsymbol{\sigma_k} - \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{R_{ij}\mu_{ij}} \tag{3.5.30}$$

3.5.2. Experiments and results

The proposed formulations are validated using a set of generated data on OTFL optimization. We used ILOG CPLEX 12.5 software to solve the model using Windows 7 platform, Intel5® Core ™ i5-3380M CPU@ 2.90GHz and 8Go of RAM.

The consideration of fixed corridors allows us to upgrade the performance of our exact method and to consider medium-sized OTLP by integrating other relevant facilities. Table 3.14 provides the notations of the test instances prefixed by ROW and gives for each instance the list of associated type of rooms.

When defining our parameters, several architectural aspects, professional advices and health care norms are to take into consideration. Thus, we do not opt for long rows in order to reduce personal's traveling distances and their dispersion, which acts on the collective functioning. OT corridors can be categorized into public corridor, sterile corridor, restricted corridor, etc. according to the type of flow going across. Therefore, facilities to make adjacent to a corridor depend on their type, their operating mode and the logic of the patient pathway.

Instance's size	Room list
ROW24	6 operating rooms, 3 induction rooms, 3 scrub room, 2 PACU, 1 decontamination, 1 sterile arsenal , 1 cleaning room, 2 bed storage, 2 monitoring and 3 recovery rooms
ROW28	6 operating rooms, 3 induction rooms, 3 scrub room, 2 PACU, 1 decontamination, 1 sterile arsenal , 1 cleaning room, 2 bed storage, 3 monitoring, 4 recovery rooms and 2 changing rooms
ROW30	6 operating rooms, 3 induction rooms, 3 scrub room, 2 PACU, 1 decontamination, 2 sterile arsenal , 1 cleaning room, 2 bed storage, 3 monitoring, 4 recovery rooms, 2 changing rooms and 1 radiology
ROW32	6 operating rooms, 3 induction rooms, 3 scrub room, 2 PACU, 1 decontamination, 2 sterile arsenal , 1 cleaning room, 2 bed storage, 3 monitoring, 4 recovery rooms, 2 changing rooms, 1 radiology, 1 secretariat and 1 reception office
ROW36	6 operating rooms, 3 induction rooms, 3 scrub room, 2 PACU, 1 decontamination, 1 sterile arsenal , 1 cleaning room, 2 bed storage, 2 monitoring, 4 recovery rooms, 2 changing rooms, 1 radiology, 1 blood test, 1 secretariat, 2 waiting rooms, 1 meeting room, 1 rest room and 1 reception office
ROW40	10 operating rooms, 5 induction rooms, 5 scrub room, 2 PACU, 1 decontamination, 2 sterile arsenal , 1 cleaning room, 2 bed storage, 3 monitoring, 4 recovery rooms, 2 changing room, 1 toilet, 1 radiology, 1 secretariat and 1 reception office

Table 3.14. Description of each multi-row instances

#	Data	Objective	Variables	Constraint	Time (sec)	Number of Iteration	%Gap
ROW24	Set 1	12763800	2305	8616	10	2171	0%
	Set 2	12632571			11	2215	0%
	Set 3	12321565			15	2156	0%
	Set 4	12025652			12	2452	0%
ROW28	Set 1	14235423	3137	11732	25	10228	0%
	Set 2	14023682			22	10052	0%
	Set 3	14753249			23	10852	0%
	Set 4	14304624			20	10003	0%
ROW30	Set 1	18523625	3601	13470	34	14974	0%
	Set 2	19000235			41	14032	0%
	Set 3	18035477			43	15074	0%
	Set 4	18920450			39	14065	0%
ROW32	Set 1	20625688	4097	14288	52	17522	0%
	Set 2	20785351			47	16032	0%
	Set 3	21003220			46	17367	0%
	Set 4	21223025			51	15235	0%
ROW36	Set 1	24498000	5185	19404	70	32356	0%
	Set 2	23958533			90	38254	0%
	Set 3	24055635			79	37241	0%
	Set 4	24622587			85	34285	0%
ROW40	Set 1	29462400	6401	23960	120	54893	0%
	Set 2	30153985			139	60152	0%
	Set 3	30023600			174	58265	0%
	Set 4	28968785			186	52356	0%

Table 3.15. Results of solving static multi-row OTLP

Table 3.15 shows the computational results for the continuous mutli-floor OTPL. The obtained results are surprising both in terms of computational time and in instances' sizes for which we obtain optimal solution. The best solution of some problems is dressed in the following figures: Figure 3.16 represents the optimal OTL with 24 facilities (i.e. ROW24) and Figure 3.17 gives the optimal OTL for ROW36 with 36 facilities.

3.5.3. Summary

This part describes a Mixed Integer Programing model for the Multi-row OTLP and motivation of this formulation. The single-row and double-row FLP has been applied to several industrial contexts, but never to hospital layout problem in general or OTLP in particular. Moreover, here we solved a multi-row formulation to cope medium sized OTLP while respecting healthcare standards limitations with the objective to minimize the total traveling cost and maximize the total closeness rating.

The computational results are more satisfactory than all other formulation, due to the reduction of the number of variables and constraints. This formulation could be extended by *considering problems with intermediate aisle structure and problems of facilities with double access (i.e. adjacent to two corridors).*

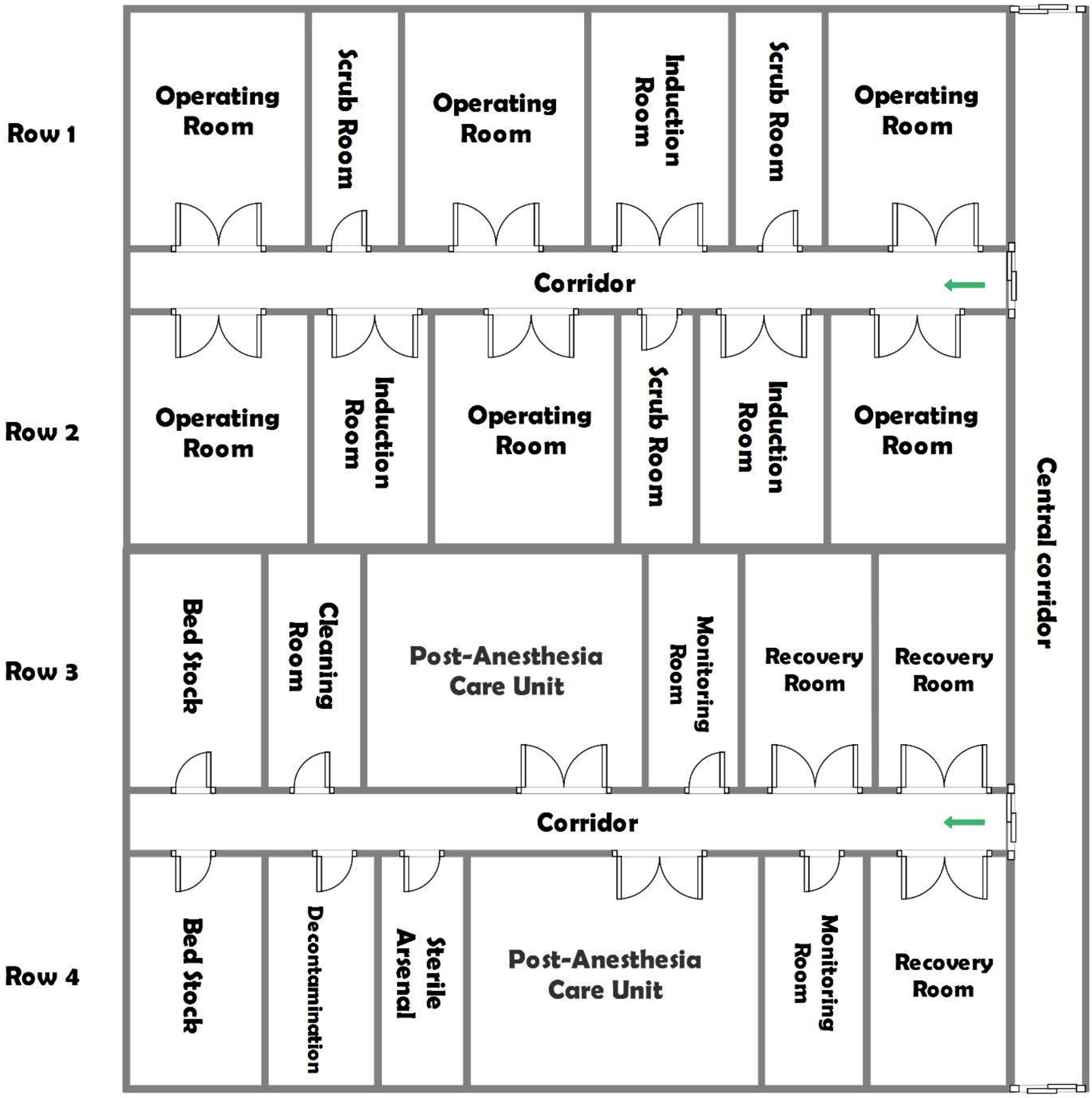


Figure 3.16. ROW24 optimal layout

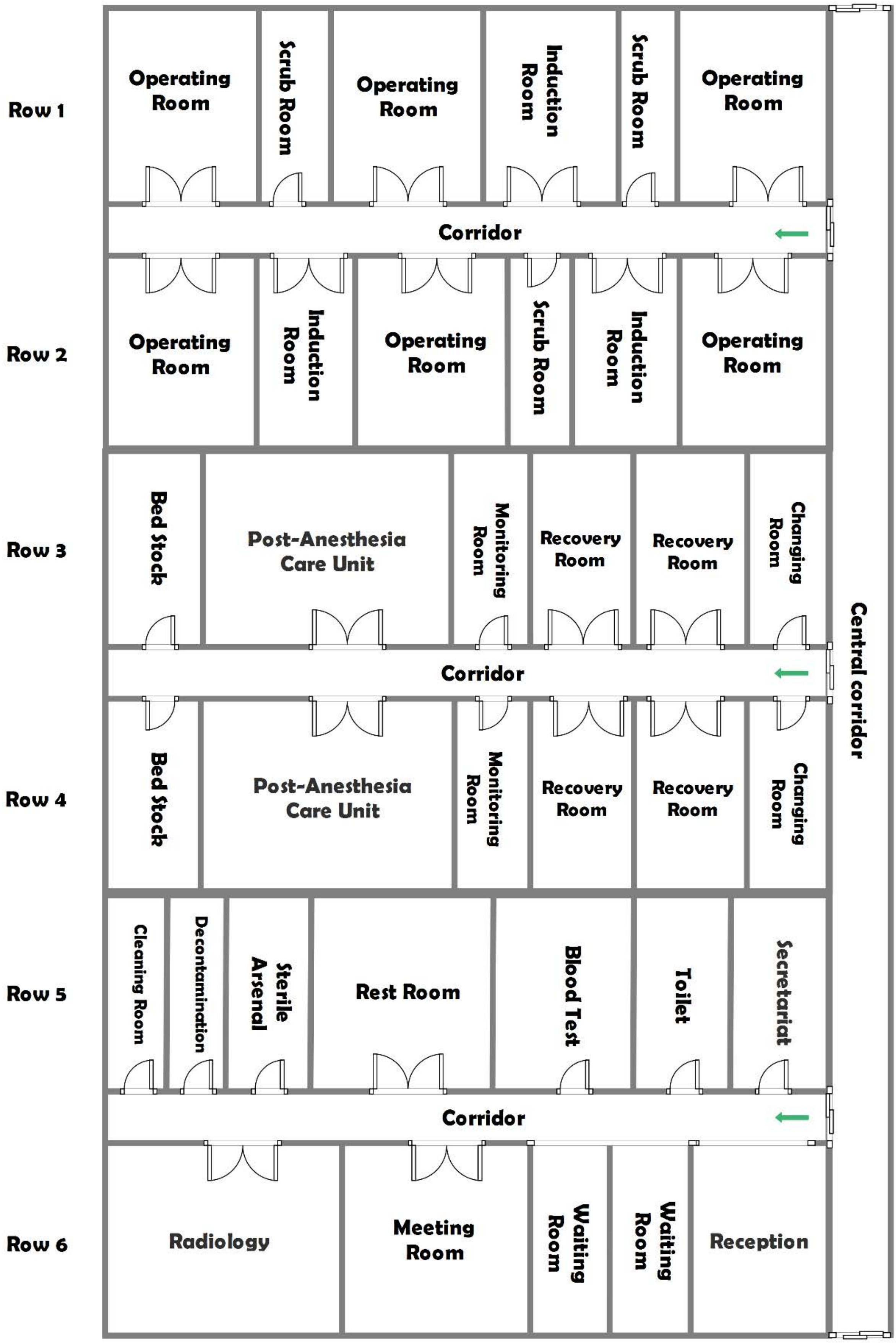


Figure 3.17. ROW36 optimal layout

3.6. Conclusion

In order to be relevant and rigorous in addressing the need of modern flexible and efficient health care systems, four models were developed for two-dimensional OT layout under Static formulation: a. simple OTLP; b. multi section OTLP; c. multi-floor OTLP and multi-row OTLP. The motivation of these formulations came from observation of realistic case of OT, the understanding of their operating mode and the comprehension of hospitals' needs.

For each formulation, a Mixed Integer Linear Programming model is developed to determine the optimal location and orientation of each facility in the final OTL while minimizing the total traveling cost and maximizing the desirable closeness rating. This is a very important advanced contribution to the field, as OTLs are planned manually without taking into account the optimization aspects. The applicability of the proposed model was demonstrated on several instances to provide optimal solutions for reasonable real sized problems.

This chapter shows the evolution of the exact method to solve Static OTLP both in terms of instances size and in terms of computational results. Several assumptions were taking into account based on real OT specifications and properties, while respecting international standards that regulate the healthcare facilities design.

In real life, OTs are not always static environments where flows between facilities and product's demands has no change or is nearly constant over time, but rather characterized by varying input data during the planning horizon. Thus, the static formulation for OTLP is not always adequate. This is why research on dynamic facility layout problem has been conducted. In the next chapter, we will focus our research on the Dynamic OTLP in order to develop a decision-making support system to seek for optimal layout in a multi-period planning.

3.7. Résumé

Pour être pertinent et rigoureux pour répondre aux besoins des systèmes de soins et de santé modernes, quatre modèles ont été développés pour le OTLP sous formulation statique : a. simple OTLP ; b. OTLP à plusieurs sections ; c. OTLP à plusieurs étages et OTLP à plusieurs rangées. La motivation de ces formulations est venue de l'observation de quelques blocs opératoires, la compréhension de leurs modes de fonctionnement et la compréhension des besoins des hôpitaux.

Pour chaque formulation, un modèle MIP est conçu pour déterminer l'emplacement et l'orientation optimale pour chaque installation dans l'OTL final tout en minimisant le coût total de déplacement et en maximisant le taux de proximité souhaitable. Ceci est une très importante contribution dans ce domaine, puisque l'OTL est généralement conçu manuellement sans

prendre en compte les aspects d'optimisation. L'applicabilité du modèle proposé a été démontrée sur plusieurs cas pour fournir des solutions optimales pour les problèmes réels de taille raisonnables.

Ce chapitre montre l'évolution de la méthode exacte pour résoudre l'OTLP statique tant en termes de tailles des instances qu'en termes de résultats de calculs. Plusieurs hypothèses ont été prises en compte sur la base de spécifications réelles et de propriétés du bloc opératoire, tout en respectant les normes internationales qui régissent la conception des établissements de soins et de santé.

Dans la vraie vie, les blocs opératoires ne sont pas toujours des environnements statiques où les flux entre les installations et les demandes de soins ne subissent aucun changement ou sont presque constants dans le temps, mais plutôt caractérisés par des données d'entrée différentes au cours de l'horizon de planification. Ainsi, la formulation statique pour l'OTLP n'est pas toujours adéquate. Voilà pourquoi la recherche sur l'OTLP dynamique a été menée. Dans le chapitre suivant, nous allons concentrer notre recherche sur l'OTLP dynamique afin de développer un système d'aide à la décision pour chercher la disposition optimale dans une planification multi-période.

Chapter IV. Dynamic Operating Theater Layout Problem: Different Models and solutions

4.1. Introduction

The continuously evolving and resource-demanding surgical facility progress, the fight against hospital infections also known as "nosocomial infections" and the continuing growth of population demand for care services are some of the evolutionary aspects of operating theaters during the past recent years that had direct impact on their conception.

The Dynamic Facility Layout Problem (DFLP) is concerned with the design of multi-period layout plans. We distinguish two types of planning horizons: fixed and rolling. In the planning period “m”, the rolling horizon consists of replacing the data at the end of period “1’”, by data from period m+1’, and it continues after finishing each period. In contrast, the fixed planning horizon just considers the first “m” period data without any replacement [Balakrishnan (2009)]).

To the best of our knowledge, Hospital layout planning is still being addressed as static FLP, where material handling flows stay invariant over a long time. As far as we know, hospitals and specifically OT are volatile. Under such conditions, some parameters such as patients demand are not stable and the facilities must be adaptive to demand change requirements. For these reasons, a static layout analysis would not be sufficient.

In order to overcome the disadvantages of the static FLP, we provide a Dynamic formulation for OTLP (DOTLP), with the objective to minimize the interdepartmental travel costs among facilities, to maximize the adjacency rating and to minimize the rearrangement costs. The DOTLP is solved by studying an individual layout for each distinctive period based on patient demands, subject to a set of constraints of distances, available areas and non-overlapping facilities according to international medical standards and specifications.

In the next chapter, we will introduce the dynamic formulation of the three last static models in chapter 3 (i.e. multi-sections OTLP, multi-floor OTLP and multi-row OTLP) and present for each dynamic formulation the new items that differentiate it from the static one while keeping the same assumptions.

4.2. Mixed Integer Programming model for Multi-Sections OTLP

In order to model the Dynamic multi-section OTLP, we provide two different approaches. The first approach is the Fixed Facility Layout Problem (FFLP), which consists of finding in one single decision a robust OT layout that is the optimal layout for all periods and that could not be rearranged in later periods. The second one is the Variable Facility Layout Problem (VFLP), which consists of generating a layout plan for each period of the planning horizon. Here, the layout will be rearranged in a way that it satisfies patient demands in each planning horizon.

The FFLP aims to minimize the traveling cost for all actors of the OT (doctors, patients, medical and non-medical staff) and maximize the total adjacency rating in each period considering varying input data (number of operations per specialty and frequency of flow for each actor) during the planning horizon. Traditional DFLP approaches consider a planning horizon, which is generally divided into periods that may be defined in weeks, months, or years where the length of the period has an impact on the final results. In fact, in short periods of time, the flows are constant and there won't be any need for layout rearrangement and the use of dynamic layout analysis may not be justified. In contrast, in long periods of time, there would be prohibitive rearrangement costs and this goes against the objective of cost minimization. This choice should be made according to the studied system, based on demand, delivered asset and service.

The FFLP approach proposes a robust layout for the various periods of the planning horizon by considering the average traveling flow between facilities. The obtained layout is then applied on all the periods despite that the flow between facilities is not the same on different periods of the planning horizon.

In the VFLP, we seek to find a layout plan for multiple periods where specialty rooms can be exchanged at the end of period in order to satisfy populations' needs in the successive periods. The objective, in addition to minimize traveling costs, is to minimize rearrangement costs generated by moving appropriated equipment for each specialty room. Table 4.1 presents a comparison between the FFLP and VFLP approaches.

	FFLP	VFLP
Facilities	Fixed facilities only	Non-movable and movable facilities
Layout	Robust layout: best possible layout for entire planning horizon;	Layout plan: adaptations in order to satisfy demand
Costs	Building fixed facilities Travelling costs	Building fixed and movable facilities Facilities rearrangement to adapt layout Travelling costs
Objective	Minimize travelling costs	Minimize travelling costs Minimize layout rearrangement costs

Table 4.1: comparison of FFLP and VFLP

4.2.1. Mathematical Formulation

Sets

- Let N= { a_i ; i=1,2,…, n} be the set of n facilities in a department,
- Let K= {e_k; k= 1… 4} be the set of k entity types: doctor, patient, medical staff or non-medical staff.
- Let S= {ε_s ; s=1, 2,…, t} be the set of sections: outer, restricted, Aseptic, disposal….
- Let C= {θ_c; c=1, 2,…, r} be the set of corridors: clean, public, soiled….
- U_i Single element set denoting the section to which facility a_i belongs
- U_c Single element set denoting the section to which corridor θ_c belongs.
- Let P= {τ_t; t=1, 2,..., p} be the set of p periods in the planning horizon;
- Let NM_p be a set of a single element denoting the non-movable facilities.

Parameters

α_i :	Length of facility a_i
βi :	Width of facility a_i
φ_{ijk}:	The movement difficulty between facility a_i to facility a_j made by an entity type e_k
l_c, d_c:	Dimensions of corridor θ_c
F_{ijkt}:	The number of trips between facility a_i to facility a_j made by an entity type e_k at a period τ_t
Fm_{ijk}:	The average flow between facility a_i to facility a_j made by an entity type e_k for the various periods in the whole planning horizon
x_{max}:	The maximum length of department;
y_{max}:	The maximum width of a department;
R_{ij}:	The desirable relationship value between facility a_i to facility a_j;
A_{ij}	Possibility of exchanging facilities affectation
C_{ij}	Relocation cost of facility a_i if shifted to a_j
ρ_1, ρ_2:	Weights for each sub-objective function.
Xl_s,Xr_s, Yb_s,Xt_s	X and Y boundary coordinates of section s

Decision Variables

Ω_{it} { 1 if length α_i of facility a_i is parallel to x-axis at a period τ_t
0 otherwise

μ_{ijt} { 1 if a_i and a_j are fully adjacent at a period τ_t
0 otherwise

V_{ijct} { 1 if traffic between facilities a_i and a_j travels through corridor θ_c at a period τ_t
0 otherwise

u_{ijt} { 1 if facility a_i and a_j are assigned to the same section at a period τ_t
0 otherwise

u_{ict} { 1 if facility a_i and corridor θ_c are assigned to the same section at a period τ_t
0 otherwise

v_{ist} { 1 if facility a_i is assigned to section ε_s at a period τ_t
0 otherwise

v_{cst} { 1 if corridor θ_c is assigned to section ε_s at a period τ_t
0 otherwise

z_{ijt}^x { 1 if facility a_i is strictly to the right of facility a_j at a period τ_t
0 otherwise

z_{ijt}^y { 1 if facility a_i is strictly above facility a_j at a period τ_t
0 otherwise

o_{ict}^{x}	{	1 if facility a_i is strictly to the right of corridor θ_c at a period τ_t 0 otherwise
o_{ict}^{y}	{	1 if facility a_i is strictly above corridor θ_c at a period τ_t 0 otherwise
x_{it} , y_{it}:		x and y coordinates of the geometric center of gravity facility a_i at a period τ_t
x_{ct} ,y_{ct} :		coordinates of the geometrical center of corridor θ_c at a period τ_t
l_{it}:		x-length of facility a_i at a period τ_t
d_{it}:		y-length of facility a_i at a period τ_t
Xp_{ijt}:		x-distance between facility a_i and a_j at a period τ_t
Yp_{ijt}:		y-distance between facility a_i and a_j at a period τ_t

i) FFLP constraints

FFLP consists on considering the average of flow between facilities for the whole planning horizon and uses equation (4.2.1) to calculate it.

$$Fm_{ijk} = \frac{\sum_{t=1}^{P} F_{ijkt}}{P} \; \forall \, i, j \neq i, k \qquad (4.2.1)$$

FFLP considers all the constraints in the static formulation of the multi-section OTPL (i.e. 3.3.1 to 3.3.48). Thus, in this dynamic formulation, the computational effort required to solve OTLP is the same as that of the static OTLP.

ii) FFLP objective function

In this model, we use a multi-objectives MIP that aims to minimize the total traveling costs and to maximize the adjacency.

$$\boldsymbol{min\ F = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{4} F_{mijk}(Xp_{ijt} + Yp_{ijt})\varphi_{ijk} * \sigma_k - \sum_{i=1}^{N} \sum_{j=1}^{N} R_{ij}\mu_{ij}} \qquad (4.2.2)$$

iii) VFLP Constraints

We will expose constraints after linearization:

$$l_{it} = \alpha_i \Omega_{it} + \beta_i (1 - \Omega_{it}) \qquad \forall i, t \qquad (4.2.3)$$

$$d_{it} = \alpha_i + \beta_i - l_{it} \qquad \forall i, t \qquad (4.2.4)$$

$$\sum_{s=1}^{S} v_{ist} = 1 \qquad \forall i, t \qquad (4.2.5)$$

$$u_{ijt} \geq v_{ist} + v_{jst} - 1 \quad \forall \, i, j \neq i, s, t \qquad (4.2.6)$$

$$u_{ijt} \leq 1 - v_{ist} + v_{jst} \quad \forall \, i, j \neq i, s, t \qquad (4.2.7)$$

$$u_{ijt} \leq 1 + v_{ist} - v_{jst} \quad \forall\, i, j \neq i, s, t \tag{4.2.8}$$

$$\sum_{c=1}^{C} V_{ijct} = 1 - u_{ijt} \quad \forall\, i, j \neq i, t \tag{4.2.9}$$

$$y_{c1t} - y_{c2t} \leq \frac{d_{c1} + d_{c2}}{2} \quad \forall\, c1, c2 \in C;\ \forall t \tag{4.2.10}$$

$$y_{c1t} - y_{c2t} \geq \frac{d_{c1} + d_{c2}}{2} \quad \forall\, c1, c2 \in C;\ \forall t \tag{4.2.11}$$

$$x_{c2t} = x_{c1t} \quad \forall\, c1, c2 \in C;\ \forall t \tag{4.2.12}$$

$$\sum_{s=1}^{S} v_{cst} = 1 \qquad \forall\, c, t \tag{4.2.13}$$

$$x_{it} - x_{jt} + X_{max}\left(1 - z_{ijt}^{x}\right) \geq \frac{l_{it} + l_{jt}}{2} \quad \forall\, i, j \neq i;\ \forall t \tag{4.2.14}$$

$$y_{it} - y_{jt} + Y_{max}\left(1 - z_{ijt}^{y}\right) \geq \frac{d_{it} + d_{jt}}{2} \quad \forall\, i, j \neq i;\ \forall t \tag{4.2.15}$$

$$z_{ijt}^{x} + z_{jit}^{x} + z_{ijt}^{y} + z_{jit}^{y} \geq 1 \ \forall\, i, j \neq i, t \tag{4.2.16}$$

$$z_{ijt}^{x} + z_{jit}^{x} \leq 1 \quad \forall\, i, j \neq i, t \tag{4.2.17}$$

$$z_{ijt}^{y} + z_{jit}^{y} \leq 1 \quad \forall\, i, j \neq i, t \tag{4.2.18}$$

$$x_{it} + \frac{l_{it}}{2} \leq Xr_s \quad \forall\, s, i \ where\ s = U_i;\ \forall t \tag{4.2.19}$$

$$y_{it} + \frac{d_{it}}{2} \leq Yt_s \quad \forall\, s, i \ where\ s = U_i;\ \forall t \tag{4.2.20}$$

$$x_{it} - \frac{l_{it}}{2} \geq Xl_s \quad \forall\, s, i \ where\ s = U_i;\ \forall t \tag{4.2.21}$$

$$y_{it} - \frac{d_{it}}{2} \geq Yb_s \quad \forall\, s, i \ where\ s = U_i;\ \forall t \tag{4.2.22}$$

$$x_{ct} + \frac{l_c}{2} \leq Xr_s \quad \forall\, s, c \ where\ s = U_c;\ \forall t \tag{4.2.23}$$

$$y_{ct} + \frac{d_c}{2} \leq Yt_s \quad \forall\, s, c \ where\ s = U_c;\ \forall t \tag{4.2.24}$$

$$x_{ct} - \frac{l_c}{2} \geq Xl_s \quad \forall\, s, c \ where\ s = U_c;\ \forall t \tag{4.2.25}$$

$$y_{ct} - \frac{d_c}{2} \geq Yb_s \quad \forall\, s, c \ where\ s = U_c;\ \forall t \tag{4.2.26}$$

$$\left(x_{ct} + \frac{l_c}{2}\right) \leq x_{it} + \frac{l_{it}}{2} + X_{max}(1 - o_{ict}^{x}) \quad (4.2.27)$$
$$\forall\, i, c \; where\; U_i = U_c;\; \forall t$$

$$\left(x_{ct} - \frac{l_c}{2}\right) + X_{max}(1 - o_{cit}^{x}) \geq x_i + \frac{l_{it}}{2} \quad (4.2.28)$$
$$\forall\, i, c \; where\; U_i = U_c;\; \forall t$$

$$\left(y_{ct} + \frac{d_c}{2}\right) \leq y_{it} + \frac{d_{it}}{2} + Y_{max}\left(1 - o_{ict}^{y}\right) \quad (4.2.29)$$
$$\forall\, i, c \; where\; U_i = U_c;\; \forall t$$

$$\left(y_{ct} - \frac{d_c}{2}\right) + Y_{max}\left(1 - o_{cit}^{y}\right) \geq y_{it} + \frac{d_{ij}}{2} \quad (4.2.30)$$
$$\forall\, i, c \; where\; U_i = U_c;\; \forall t$$

$$o_{ict}^{x} + o_{cit}^{x} + o_{ict}^{y} + o_{cit}^{y} \geq 1 \quad (4.2.31)$$
$$\forall\, i, c \; where\; U_i = U_c;\; \forall t$$

$$Xp_{ijt} \geq (x_{it} - x_{ct}) + \left(x_{ct} - x_{jt}\right) \quad (4.2.32)$$
$$\forall\, i, c \; where\; U_i = U_c;\; \forall t$$

$$Xp_{ijt} \geq (x_{it} - x_{ct}) + \left(x_{jt} - x_{ct}\right) \quad (4.2.33)$$
$$\forall\, i, c \; where\; U_i = U_c;\; \forall t$$

$$Xp_{ijt} \geq (x_{ct} - x_{it}) + \left(x_{ct} - x_{jt}\right) \quad (4.2.34)$$
$$\forall\, i, c \; where\; U_i = U_c;\; \forall t$$

$$Xp_{ijt} \geq (x_{ct} - x_{it}) + \left(x_{jt} - x_{ct}\right) \quad (4.2.35)$$
$$\forall\, i, c \; where\; U_i = U_c;\; \forall t$$

$$Yp_{ijt} \geq \left(y_{it} - y_{jt}\right) \;\; \forall\, i, j \neq i, t \quad (4.2.36)$$

$$Yp_{ijt} \geq \left(y_{jt} - y_{it}\right) \;\; \forall\, i, j \neq it \quad (4.2.37)$$

Constraints (4.2.2) to (4.2.37) are the same as constraints (3.3.1) to (3.3.48) in chapter 3 with specification of the period *t*.

$$x_{it} = x_{it-1} \quad \forall\, i \in NM_p\; \forall\, t\; \in T \quad (4.2.38)$$
$$y_{it} = y_{it-1} \quad \forall\, i \in NM_p\; \forall\, t\; \in T \quad (4.2.39)$$

VFLP consists of interchanging the positions of some facilities from a period to another. Thus, constraints (4.2.38) and (4.2.39) insure that the non-movable facilities do not move during the planning horizon.

iv) VFLP objective function

$$\min F = \sum_{t=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{4} F_{ijkt}(Xp_{ijt} + Yp_{ijt})(\varphi_{ijk} * \sigma_k) - \sum_{t=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} R_{ij}\mu_{ijt} + \sum_{t=2}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}C_{ij}r_{tij} \tag{4.2.40}$$

The main objectives are minimizing the traveling costs and maximizing the adjacency rating for each period in the planning horizon; and the minimizing of rearrangement costs when reallocating movable facilities.

4.2.2. Experiments and results

The proposed formulations are validated using a set of generated data on OTFL optimization. We used ILOG CPLEX 12.5 software to solve the model using Windows 7 platform, Intel5® Core ™ i5-3380M CPU@ 2.90GHz and 8Go of RAM. For computational study, we generate four sets of data for each instance size. We consider two different planning horizons for three and six periods, while a period refer to one month of flows.

Finding an optimal OTFL to minimize the various costs requires the following specification of requirements: the number of sections, the land area required by the facility, the number of corridors within the facility, the length, width and orientation of each facility, the facility and corridors allocations to each section, the duration of the planning horizon and the number of periods. Table 4.2 explains the notation of the test instances prefixed by SEC and gives for each one the list of selected rooms where P3 and P6 refer to a planning horizon of 3 periods and 6 periods, respectively.

Instance's size	Description	Room list
SEC11_P3 **SEC11_P6**	An instance of eleven facilities with a planning horizon of three and six periods	4 operating rooms, 1 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room and 2 corridors
SEC13_P3 **SEC13_P6**	An instance of thirteen facilities with a planning horizon of three and six periods	4 operating rooms, 2 induction rooms, 1 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room and 2 corridors
SEC16_P3 **SEC16_P6**	An instance of sixteen facilities with a planning horizon of three and six periods	4 operating rooms, 2 induction rooms, 2 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room, 1 changing room, 1 rest room and 2 corridors
SEC17_P3 **SEC17_P6**	An instance of seventeen facilities with a planning horizon of three and six periods	4 operating rooms, 2 induction rooms, 2 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room, 2 changing room, 1 bed storage and 2 corridors

Table 4.2. Description of each multi-section instances

i) FFLP results

As announced above, we consider a planning horizon with three and six periods for different FFLP instances. Each set of data in test instances was solved to optimality in reasonable time for SEC11, SEC13 and SEC16. The computational time increases while the number of facilities increases until eighteen, for which optimality was not attained. We can also observe the variation of computational time between sets of data in the same instance, which is due to the generated data that can be complex and for which the algorithm needs more iterations to find a better solution; or can be plain and for which the algorithm needs less time to solve. For each instance, we select the best objective to minimize the traveling costs for FFLP and we address the OT layout using commercial software.

From Table 4.3 we can observe that the complexity of the robust model is the same as that of the static model, according to the number of variables and constraints. As for the problem size, we solved to the optimum until we reached seventeen facilities in a time below two hours. Figure 4.1 and Figure 4.2 show the optimal layout for SEC11_P3 and SEC16_P3 as example for a choosing data set.

#	Data	Objective	Variables	Constraint	Time (sec)	Number of Iteration	Gap
SEC11_P3	Set 1	287950	657	1264	7.05	119542	0%
	Set 2	289950			5.69	135639	0%
	Set 3	298350			1.19	30270	0%
	Set 4	278350			1.22	41431	0%
SEC11_P6	Set 1	275900			5.35	109421	0%
	Set 2	267690			4.27	128036	0%
	Set 3	296700			1.37	33996	0%
	Set 4	283700			1.33	42214	0%
SEC13_P3	Set 1	333050	955	1872	26.04	756721	0%
	Set 2	333450			30.20	949028	0%
	Set 3	321200			11.44	329784	0%
	Set 4	311250			17.89	485932	0%
SEC13_P6	Set 1	366100			26.15	655638	0%
	Set 2	376300			32.46	862929	0%
	Set 3	382400			11.00	382054	0%
	Set 4	382600			14.02	437654	0%
SEC16_P3	Set 1	416150	1507	3009	749.21	20415301	0%
	Set 2	428100			672.16	13843465	0%
	Set 3	404450			894.96	21968228	0%
	Set 4	403430			784.44	16747586	0%

SEC16_P6	Set 1	478300			1055.57	22471650	0%
	Set 2	458360			459.67	10237416	0%
	Set 3	481950			717.29	15872121	0%
	Set 4	471900			541.81	14496460	0%
SEC17_P3	set 1	900050	1719	3448	2843.68	52523486	0%
	set 2	909050			2837.24	48858039	0%
	set 3	914450			4470.70	78473936	0%
	set 4	912350			3935.89	66785270	0%
SEC17_P6	set 1	986050			3798.69	65853015	0%
	set 2	987550			5197.13	91055295	0%
	set 3	973250			3681.09	73532298	0%
	set 4	972400			3107.48	60423216	0%

Table 4.3. Results of solving Dynamic multi-sections OTLP with FFLP

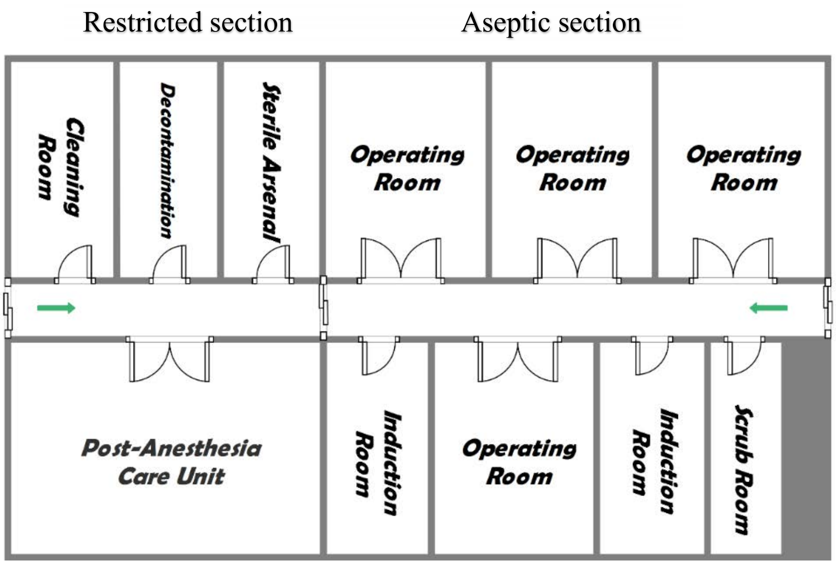


Figure 4.1. Robust layout of FFLP for SEC13_P3

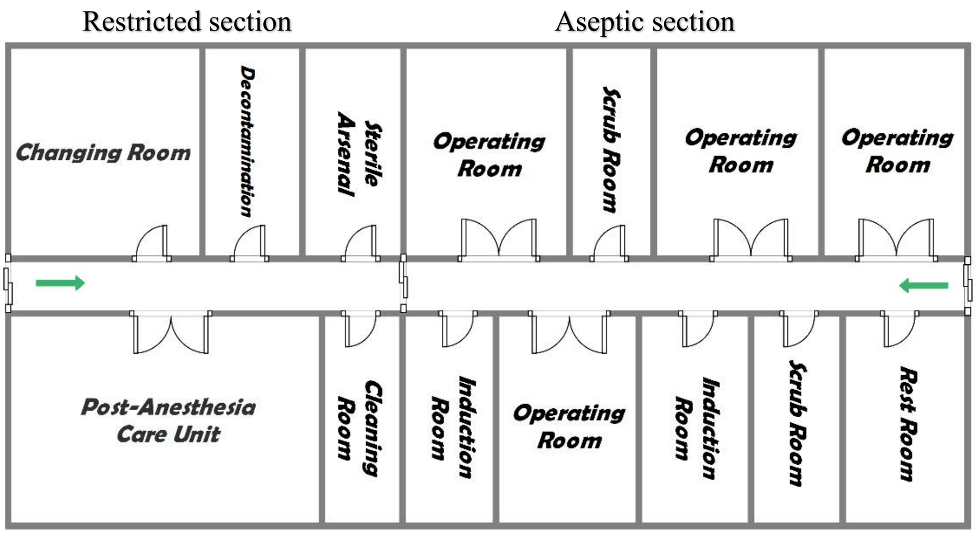


Figure 4.2. Robust layout of FFLP for SEC16_P3

ii) VFLP results

In our VFLP approach, the facilities that can be rearranged are the Operating Rooms and Induction Rooms in a way that two ORs can exchange their specialty and move the needed equipment and materials from one to the other to satisfy patients demand. We adopt the same assumption for induction rooms even if an induction room is a standards facility, but it still holds the same equipment required for similar surgical operations. Rearrangement of induction rooms assumes the exchange of their equipment. Assumptions used in this formulation were approved by a senior executive of the surgical staff in Roanne's Hospital.

In this work, we consider a multi-disciplinary OT, which means that each operating room is affected to a surgical specialty. To rearrange the OT layout, we adopt the specialty exchanging between operating rooms, since restructuration of the whole OT after each period can be very expensive. In this case, the OT will keep the first layout configuration and room specialty could be exchanged at the end of each period to satisfy patient demands.

#	Data	Objective	Variables	Constraint	Time (sec)	Number of Iteration	Gap	Moves
SEC11_P3	Set 1	8368750	1952	3302	354	2511079	0%	8
	Set 2	8376700			432	1627011	0%	8
	Set 3	8384350			242	1372466	0%	6
	Set 4	8382200			200	1244579	0%	6
SEC13_P3	Set 1	14527450	2866	4886	1566	32343456	1.14%	10
	Set 2	14884850			1166	30323430	0.40%	10
	Set 3	14793200			2024	52483377	0%	8
	Set 4	14460520			2066	52523486	0%	8
SEC16_P3	Set 1	22887850	4567	7847	5395	217349500	4.11%	10
	Set 2	23602650			4380	215346500	3.16%	8
	Set 3	23023350			11274	263999558	2.53%	10
	Set 4	22357990			12351	303009506	1.40%	6

Table 4.4. Results of solving Dynamic multi-sections OTLP with VFLP

Table 4.4 shows the computational results of Dynamic OTLP with VFLP approach. The column moves indicate the needed exchanges between movable facilities in the planning horizon (i.e. when two facilities exchange their specialty moves =2). For each instance, we select the optimal solution from the data sets and we address the OT layout using commercial software. Figure 4.3 and Figure 4.4 show the optimal layout for INS11_P6 and INS16_P3 respectively in each period of the planning horizon. For experimentation, we consider the four surgery specialties (e.g. OR1= general surgery, OR2= orthopedic surgery, OR3= Gynecology and OR4= Urology).

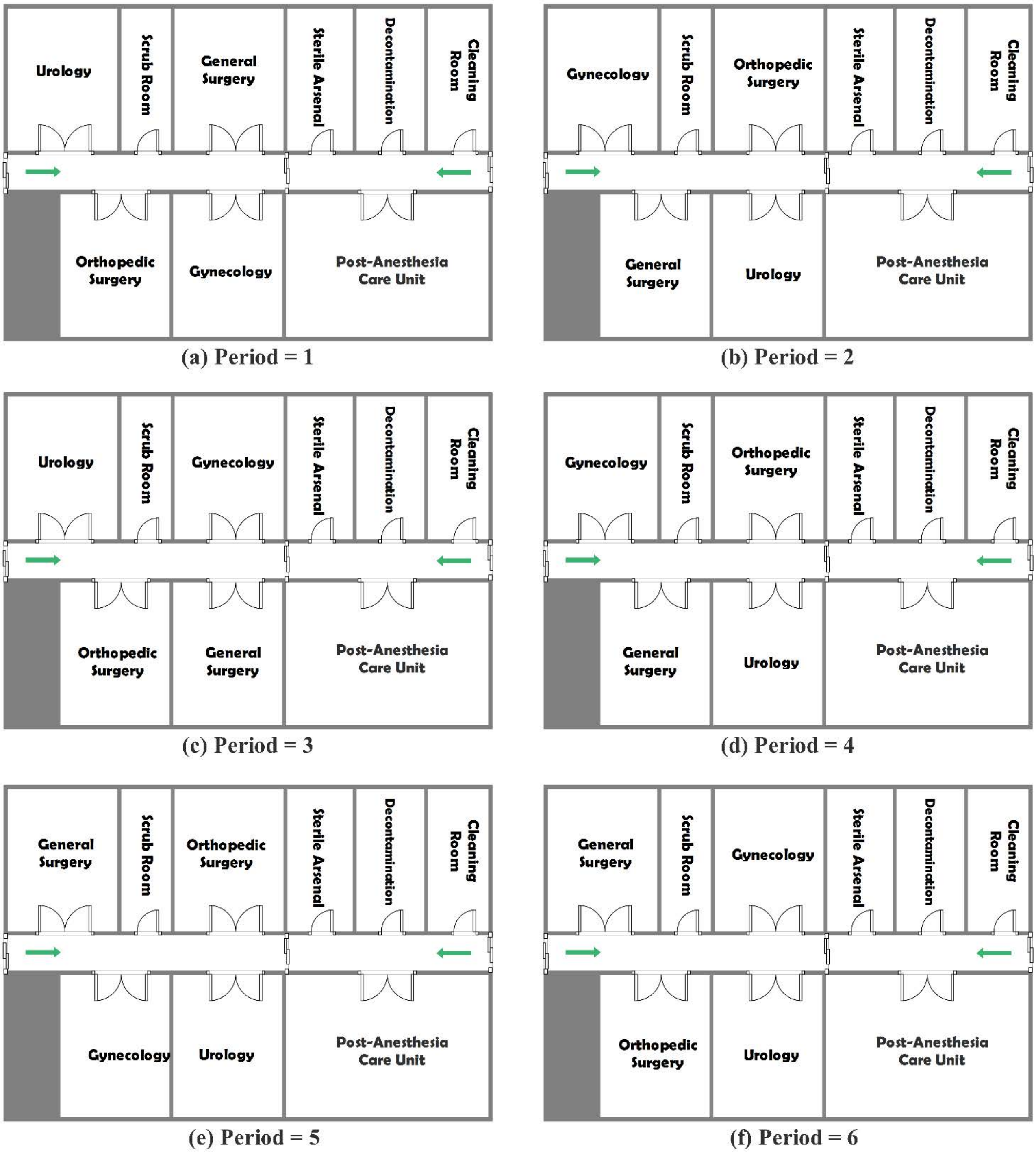


Figure 4.3. VFLP layout for SEC11_P6 for planning horizon of six periods

From the obtained layouts for each instance size, we observe the reallocation of facilities within the OT at different periods to ensure optimality and adequacy for various surgical demands. These assumptions and results were viewed and approved by a senior manager of the OT of Roanne's hospital.

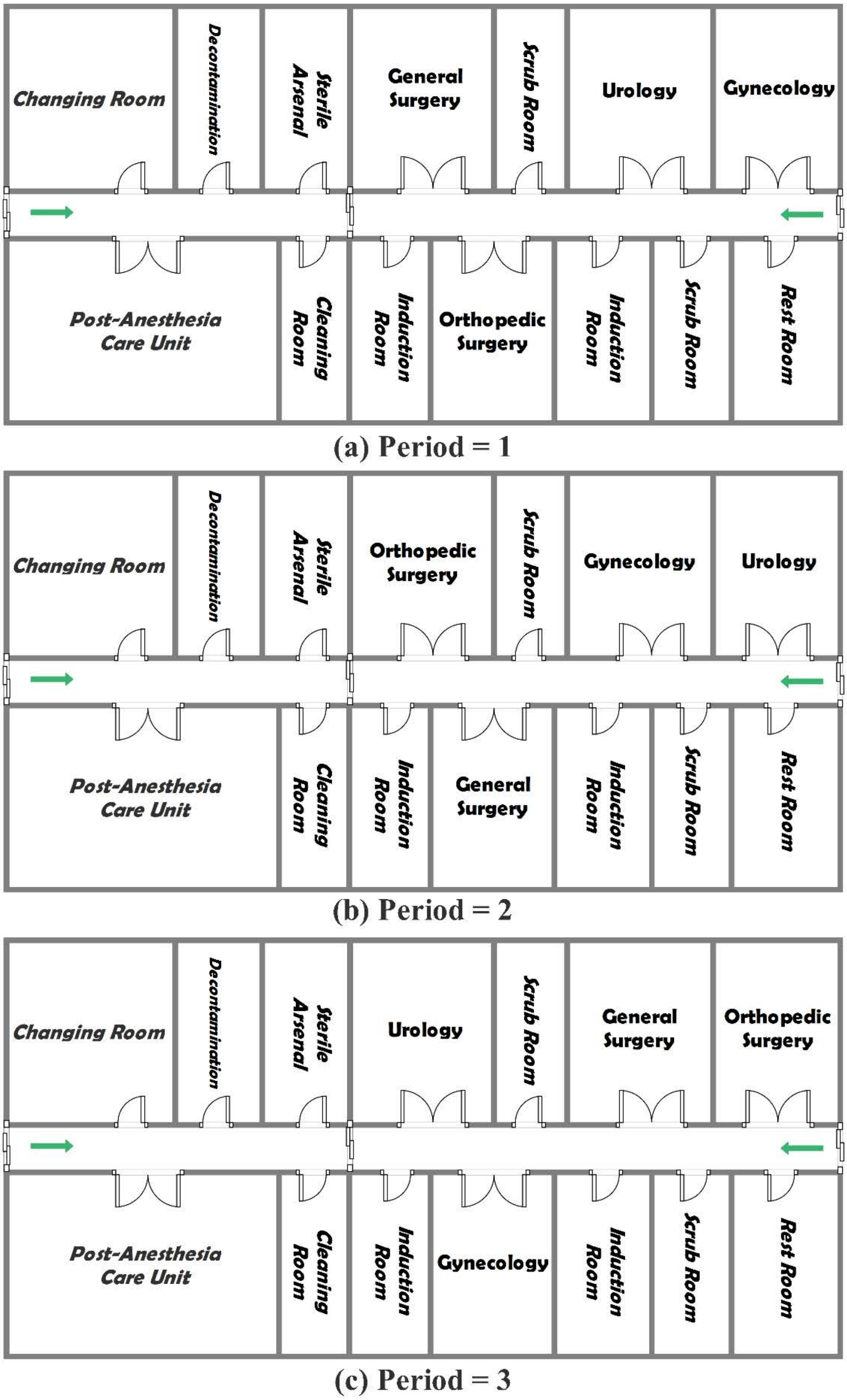


Figure 4.4. VFLP layout for SEC16_P3 for planning horizon of three periods

4.2.3. Summary

This part describes two Mixed Integer Programing models for the Dynamic Multi-Section OTLP and motivation of these formulations. The FFLP consists to find in only one decision a *robust OT layout for the whole planning horizon while minimizing the total traveling cost and maximizing the total closeness rating. In contrast, the VFLP seeks to generate a layout plan for each period in the planning horizon while minimizing the total traveling cost, maximizing the total closeness rating and minimizing the total reallocation costs.*

The computational results were optimal for FFLP in reasonable time, while the designed layouts were satisfying international standards in term of safety and hygiene. For VFLP we did not attain optimality for bigger instance sizes, but final OT layouts show that even the remaining gap, solution is near to the exact proposed static OTLP solution.

To deal with other cases of real OT structure, next chapter will introduce the Dynamic formulation for the multi-floor OTLP.

4.3. Mixed Integer Programming model for Multi-Floors OTLP

To return to the screening static multi-floor formulation used above, the dynamic formulation keeps the same assumptions and constraints. We propose two approaches to solve this formulation (i.e. FFLP and VFLP) by considering fixed and movable facilities in multiple periods of the planning horizon.

4.3.1. Mathematical Formulation

Sets

In addition to the sets of the static multi-floor OTLP (cf. §3.4.1), we consider:

- P= $\{\tau_t$; t=1, 2,..., $p\}$ be the set of p periods in the planning horizon;
- NM_p be a set of a single element denoting the non-movable facilities.

Parameters

The new parameters in this formulation are:

F_{ijkt}:	The number of trips between facility a_i to facility a_j made by an entity type e_k at a period τ_t
Fm_{ijk}:	The average flow between facility a_i to facility a_j made by an entity type e_k for the various periods in the whole planning horizon
A_{ij}	Possibility of exchanging facilities affectation
C_{ij}	Relocation cost of facility a_i if shifted to a_j

Decision Variables

$$\Omega_{it} \quad \begin{cases} 1 \text{ if length } \alpha_i \text{ of facility } a_i \text{ is parallel to x-axis at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$$

μ_{ijt} $\begin{cases} 1 \text{ if } a_i \text{ and } a_j \text{ are fully adjacent at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

V_{ijet} $\begin{cases} 1 \text{ if traffic between facilities } a_i \text{ and } a_j \text{ travels through elevator } \vartheta_e \text{ at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

t_{ijt} $\begin{cases} 1 \text{ if facility } a_i \text{ and } a_j \text{ are assigned to the same floor at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

t_{ict} $\begin{cases} 1 \text{ if facility } a_i \text{ and corridor } \theta_c \text{ are assigned to the same floor at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

v_{ift} $\begin{cases} 1 \text{ if facility } a_i \text{ is assigned to floor } \varepsilon_f \text{ at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

v_{cft} $\begin{cases} 1 \text{ if corridor } \theta_c \text{ is assigned to floor } \varepsilon_f \text{ at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

v_{eft} $\begin{cases} 1 \text{ if elevator } \vartheta_e \text{ is assigned to floor } \varepsilon_f \text{ at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

z_{ijt}^{x} $\begin{cases} 1 \text{ if facility } a_i \text{ is strictly to the right of facility } a_j \text{ at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

z_{ijt}^{y} $\begin{cases} 1 \text{ if facility } a_i \text{ is strictly above facility } a_j \text{ at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

o_{ict}^{x} $\begin{cases} 1 \text{ if facility } a_i \text{ is strictly to the right of corridor } \theta_c \text{ at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

o_{ict}^{y} $\begin{cases} 1 \text{ if facility } a_i \text{ is strictly above corridor } \theta_c \text{ at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

ω_{ift}^{x} $\begin{cases} 1 \text{ if facility } a_i \text{ is strictly to the right of elevator } \vartheta_e \text{ at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

ω_{cft}^{y} $\begin{cases} 1 \text{ if elevator } \vartheta_e \text{ is strictly above corridor } \theta_c \text{ at a period } \tau_t \\ 0 \text{ otherwise} \end{cases}$

x_{it} , y_{it}: x and y coordinates of the geometric center of gravity facility a_i at a period τ_t

x_{ct} ,y_{ct} : coordinates of the geometrical center of corridor θ_c at a period τ_t

x_{et} ,y_{et} : coordinates of the geometrical center of elevator ϑ_e at a period τ_t

l_{it}: x-length of facility a_i at a period τ_t

d_{it}: y-length of facility a_i at a period τ_t

Xp_{ijt}: x-distance between facility a_i and a_j in the same floor at a period τ_t

Yp_{ijt}: y-distance between facility a_i and a_j in the same floor at a period τ_t

Xe_{iet}: x-distance between facility a_i and elevator ϑ_e at a period τ_t

Ye_{iet}: y-distance between facility a_i and elevator ϑ_e at a period τ_t

Zp_{ijt}: Vertical distance between facility a_i and a_j in different floors at a period τ_t

i) FFLP Constraints

FFLP considers all the constraints in the static formulation of the multi-floor OTPL (i.e. 3.4.1 to 3.4.58). Equation (4.3.1) computes the average flow frequency in the various periods of the planning horizon.

$$Fm_{ijk} = \frac{\sum_{t=1}^{P} F_{ijkt}}{P} \; \forall\, i, j \neq i, k \tag{4.3.1}$$

ii) FFLP objective function

In this model, we use a multi-objective MIP that aims to minimize the total traveling costs and to maximize the adjacency by considering vertical distance and cost factor relative to the elevator use.

$$\begin{aligned} \boldsymbol{min\ F} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{4} \boldsymbol{Fm_{ijk}}\left[(\boldsymbol{Xp_{ij}} + \boldsymbol{Yp_{ij}})\boldsymbol{\varphi_{ijk}} * \boldsymbol{\sigma_k} + \boldsymbol{Ze_{ij}\sigma_e}\right] \\ - \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{R_{ij}\mu_{ij}} \end{aligned} \tag{4.3.2}$$

iii) VFLP constraints

Equations (4.3.3) to (4.3.60) insure constraints of orientation, of non-overlapping, of area and constraints to calculate distances in each period of the planning horizon.

$$l_{it} = \alpha_i \Omega_{it} + \beta_i (1 - \Omega_{it}) \qquad \forall i, t \tag{4.3.3}$$

$$d_{it} = \alpha_i + \beta_i - l_{it} \qquad \forall i, t \tag{4.3.4}$$

$$\sum_{f=1}^{F} v_{ift} = 1 \qquad \forall i\,, t \tag{4.3.5}$$

$$t_{ijt} = \left| v_{ift} + v_{jft} - 1 \right| \; \forall\, i, j \neq i, f, t \tag{4.3.6}$$

$$t_{ijt} \geq v_{ift} + v_{jft} - 1 \;\; \forall\, i, j \neq i, f, t \tag{4.3.7}$$

$$t_{ijt} \leq 1 - v_{ift} + v_{jft} \;\; \forall\, i, j \neq i, f, t \tag{4.3.8}$$

$$t_{ijt} \leq 1 + v_{ift} - v_{jft} \quad \forall\, i, j \neq i, f, t \tag{4.3.9}$$

$$\sum_{f=1}^{F} v_{cft} = 1 \qquad \forall i, t \tag{4.3.10}$$

$$t_{ict} \geq v_{ift} + v_{cft} - 1 \quad \forall\, i, c, f, t \tag{4.3.11}$$

$$t_{ict} \leq 1 - v_{ift} + v_{cft} \quad \forall\, i, c, f, t \tag{4.3.12}$$

$$t_{ict} \leq 1 + v_{ift} - v_{cft} \quad \forall\, i, c, f, t \tag{4.3.13}$$

$$\sum_{e=1}^{E} V_{ijet} = 1 - t_{ijt} \quad \forall\, i, j \neq i \tag{4.3.14}$$

$$\begin{gathered} x_{e1t} = x_{e2t} \\ \forall\, e1 = 1, \dots, E-1;\; e2 = e1+1, \dots, E \end{gathered} \tag{4.3.15}$$

$$\begin{gathered} y_{e1t} = y_{e2t} \\ \forall\, e1 = 1, \dots, E-1;\; e2 = e1+1, \dots, E \end{gathered} \tag{4.3.16}$$

$$\sum_{e=1}^{E} v_{eft} = 1 \qquad \forall\, f, t \tag{4.3.17}$$

$$x_{it} - x_{jt} + X_{max}\left(2 - z_{ijt}^{x} - t_{ijt}\right) \geq \frac{l_{it} + l_{jt}}{2} \quad \forall\, i, j \neq i, t \tag{4.3.18}$$

$$y_{it} - y_{jt} + Y_{max}\left(2 - z_{ijt}^{y} - t_{ijt}\right) \geq \frac{d_{it} + d_{jt}}{2} \quad \forall\, i, j \neq i, t \tag{4.3.19}$$

$$z_{ijt}^{x} + z_{jit}^{x} + z_{ijt}^{y} + z_{jit}^{y} \geq t_{ijt} \quad \forall\, i, j \neq i, t \tag{4.3.20}$$

$$z_{ijt}^{x} + z_{jit}^{x} \leq t_{ijt} \quad \forall\, i, j \neq i, t \tag{4.3.21}$$

$$z_{ijt}^{y} + z_{jit}^{y} \leq t_{ijt} \quad \forall\, i, j \neq i, t \tag{4.3.22}$$

$$x_{it} + \frac{l_{it}}{2} \leq X_{max} \quad \forall\, i, t \tag{4.3.23}$$

$$y_{it} + \frac{d_{it}}{2} \leq Y_{max} \quad \forall\, i, t \tag{4.3.24}$$

$$x_{it} - \frac{l_{it}}{2} \geq 0 \quad \forall\, i, t \tag{4.3.25}$$

$$y_{it} - \frac{d_{it}}{2} \geq 0 \quad \forall\, i, t \tag{4.3.26}$$

$$x_{ct} + \frac{l_{c}}{2} \leq X_{max} \quad \forall\, c, t \tag{4.3.27}$$

$$y_{ct} + \frac{d_{c}}{2} \leq Y_{max} \quad \forall\, c, t \tag{4.3.28}$$

$$x_{ct} - \frac{l_{c}}{2} \geq 0 \quad \forall\, c, t \tag{4.3.29}$$

$$y_{ct} - \frac{d_{c}}{2} \geq 0 \quad \forall\, c, t \tag{4.3.30}$$

$$x_{et} + \frac{l_e}{2} \leq X_{max} \quad \forall\, e, t \tag{4.3.31}$$

$$y_{et} + \frac{d_e}{2} \leq Y_{max} \quad \forall\, e, t \tag{4.3.32}$$

$$x_{et} - \frac{l_e}{2} \geq 0 \quad \forall\, e, t \tag{4.3.33}$$

$$y_{et} - \frac{d_e}{2} \geq 0 \quad \forall\, e, t \tag{4.3.34}$$

$$\left(x_{ct} + \frac{l_c}{2}\right) \leq x_{it} + \frac{l_{it}}{2} + X_{max}(2 - o^x_{ict} - t_{ict}) \ \forall\, i, c, t \tag{4.3.35}$$

$$\left(x_{ct} - \frac{l_c}{2}\right) + X_{max}(2 - o^x_{cit} - t_{ict}) \geq x_{it} + \frac{l_{it}}{2} \quad \forall\, i, c, t \tag{4.3.36}$$

$$\left(y_{ct} + \frac{d_c}{2}\right) \leq y_{it} + \frac{d_{it}}{2} + Y_{max}\left(2 - o^y_{ict} - t_{ict}\right) \ \forall\, i, c\,, t \tag{4.3.37}$$

$$\left(y_{ct} - \frac{d_c}{2}\right) + Y_{max}\left(2 - o^y_{cit} - t_{ict}\right) \geq y_{it} + \frac{d_{it}}{2} \quad \forall\, i, c\,, t \tag{4.3.38}$$

$$o^x_{ict} + o^x_{cit} + o^y_{ict} + o^y_{cit} \geq t_{ict} \ \ \forall\, i, c, t \tag{4.3.39}$$

$$x_{it} - x_{et} + X_{max}(1 - \omega^x_{iet}) \geq \frac{l_{it} + l_e}{2} \quad \forall\, i, e, t \tag{4.3.40}$$

$$y_{it} - y_{et} + Y_{max}\left(1 - \omega^y_{iet}\right) \geq \frac{d_{it} + d_e}{2} \quad \forall\, i, e, t \tag{4.3.41}$$

$$\omega^x_{iet} + \omega^x_{eit} + \omega^y_{iet} + \omega^y_{eit} \geq 1 \quad \forall\, i, e, t \tag{4.3.42}$$

$$\omega^x_{iet} + \omega^x_{eit} \leq 1 \quad \forall\, i, e, t \tag{4.3.43}$$

$$\omega^y_{iet} + \omega^y_{eit} \leq 1 \quad \forall\, i, e, t \tag{4.3.44}$$

$$Yp_{ijt} \geq (y_{it} - y_{ct}) + \left(y_{ct} - y_{jt}\right) - (1 - t_{ict})(X_{max} + Y_{max}) \\ \forall\, i, j \neq i, c, t \tag{4.3.45}$$

$$Yp_{ijt} \geq (y_{it} - y_{ct}) + \left(y_{jt} - y_{ct}\right) - (1 - t_{ict})(X_{max} + Y_{max}) \\ \forall\, i, j \neq i, c, t \tag{4.3.46}$$

$$Yp_{ijt} \geq (y_{ct} - y_{it}) + \left(y_{ct} - y_{jt}\right) - (1 - t_{ict})(X_{max} + Y_{max}) \\ \forall\, i, j \neq i, c, t \tag{4.3.47}$$

$$Yp_{ijt} \geq (y_{ct} - y_{it}) + \left(y_{jt} - y_{ct}\right) - (1 - t_{ict})(X_{max} + Y_{max}) \\ \forall\, i, j \neq i, c, t \tag{4.3.48}$$

$$Xp_{ijt} \geq \left(x_{it} - x_{jt}\right) - (1 - t_{ijt})(X_{max} + Y_{max}) \\ \forall\, i, j \neq i, t \tag{4.3.49}$$

$$Xp_{ijt} \geq \left(x_{jt} - x_{it}\right) - (1 - t_{ijt})(X_{max} + Y_{max}) \\ \forall\, i, j \neq i, t \tag{4.3.50}$$

$$Ye_{iet} \geq (y_{it} - y_{ct}) + (y_{ct} - y_{et}) \ \forall\, i, e, c, t \tag{4.3.51}$$

$$Ye_{iet} \geq (y_{it} - y_{ct}) + (y_{et} - y_{ct}) \ \forall\, i, e, c, t \tag{4.3.52}$$

$$Ye_{iet} \geq (y_{ct} - y_{it}) + (y_{ct} - y_{et}) \; \forall \, i, e, c, t \tag{4.3.53}$$

$$Ye_{iet} \geq (y_{ct} - y_{it}) + (y_{et} - y_{ct}) \; \forall \, i, e, c, t \tag{4.3.54}$$

$$Xe_{iet} \geq (x_{it} - x_{et}) \; \forall \, i, e, t \tag{4.3.55}$$

$$Xe_{iet} \geq (x_{et} - x_{it}) \; \forall \, i, e, t \tag{4.3.56}$$

$$Ze_{ijt} \geq H \sum_{b=1}^{F} b\left(v_{ift} - v_{jft}\right) \; \forall \, i, j \neq i, f, t \tag{4.3.57}$$

$$Ze_{iet} \geq H \sum_{b=1}^{F} b\left(v_{jft} - v_{ift}\right) \; \forall \, i, j \neq i, f, t \tag{4.3.58}$$

$$Yp_{ijt} \geq Ye_{iet} + Ye_{jet} - [t_{ijt}Y_{max} + \left(1 - V_{ijet}\right)Y_{max}] \quad \forall \, i, j, e, t \tag{4.3.59}$$

$$Xp_{ijt} \geq Xe_{iet} + Xe_{jet} - [t_{ijt}X_{max} + \left(1 - V_{ijet}\right)X_{max}] \quad \forall \, i, j, e, t \tag{4.3.60}$$

$$x_{it} = x_{it-1} \quad \forall \, i \in NM_p \; \forall \, t \in T \tag{4.3.61}$$

$$y_{ti} = y_{it-1} \quad \forall \, i \in NM_p \; \forall \, t \in T \tag{4.3.62}$$

$$v_{ift} = v_{ift-1} \; \forall i, f, t \tag{4.3.63}$$

To permit exchanging the positions of some facilities from a period to another, VFLP uses constraints (4.3.61) and (4.3.62) to ensure that the non-movable facilities do not move during the planning horizon, while constraint (4.3.63) ensures that facilities remain affected to the same floor all over the planning horizon.

iv) VFLP objective function

$$\boldsymbol{min\ F} = \sum_{t=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{4} \boldsymbol{F_{ijkt}}\left[(\boldsymbol{Xp_{ijt}} + \boldsymbol{Yp_{ijt}})\boldsymbol{\varphi_{ijk}} * \boldsymbol{\sigma_k} + \boldsymbol{Ze_{ijt}\sigma_e}\right] - \sum_{t=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{R_{ij}\mu_{ijt}} + \sum_{t=2}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{A_{ij}C_{ij}r_{tij}} \tag{4.3.64}$$

The main objectives are minimizing the traveling costs and maximizing the adjacency rating for each period in the planning horizon; and the minimizing of rearrangement costs when reallocating movable facilities.

4.3.2. Experiments and results

The proposed formulations are validated using a set of generated data on OTFL optimization. We used ILOG CPLEX 12.5 software to solve the model using Windows 7 platform, Intel5® Core ™ i5-3380M CPU@ 2.90GHz and 8Go of RAM. For a computational study, we generate four sets of data for each instance size. We consider two different planning horizons for three and six periods.

Instance's size		Room list
FLR12_P3 **FLR12_P6**	An instance of twelve facilities with a planning horizon of three and six periods	3 operating rooms, 1 scrub room, 1 PACU, 1 sterile arsenal , 1 monitoring, 1 recovery rooms, 2 corridors and 2 elevators
FLR14_P3 **FLR14_P6**	An instance of fourteen facilities with a planning horizon of three and six periods	4 operating rooms, 1 induction rooms, 1 scrub room, 1 PACU, 1 sterile arsenal , 1 monitoring, 1 recovery rooms, 2 corridors and 2 elevators
FLR16_P3 **FLR16_P6**	An instance of sixteen facilities with a planning horizon of three and six periods	4 operating rooms, 1 induction rooms, 1 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal , 1 monitoring, 2 recovery rooms, 2 corridors and 2 elevators
FLR18_P3 **FLR18_P6**	An instance of eighteen facilities with a planning horizon of three and six periods	4 operating rooms, induction rooms, 1 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room, 1 bed storage, 1 monitoring, 2 recovery rooms, 2 corridors and 2 elevators
FLR20_P3 **FLR20_P6**	An instance of twenty facilities with a planning horizon of three and six periods	4 operating rooms, 2 induction rooms, 2 scrub room, 1 PACU, 1 decontamination, 1 sterile arsenal, 1 cleaning room, 1 bed storage, 1 monitoring, 2 recovery rooms, 2 corridors and 2 elevators

Table 4.5. Description of each multi-floor instances

Table 4.5 provides the notations of the test instances prefixed by FLR and assigns for each one the list of selected rooms where P3 and P6 refer to a planning horizon of 3 periods and 6 periods, respectively.

To illustrate the two formulations, we perform several numerical experiments in this section. We generated data (i.e. flow frequency, cost factors and desirable relationship) for different instance sizes based on real life observations, and set the maximum number of floors to two and the maximum number of elevators to one; and we solved each instance using four different scenarios.

i) FFLP results

#	Data	Objective	Variables	Constraint	Time (sec)	Number of Iteration	%Gap
FLR12_P3	Set 1	4235482	2580	4806	2555	34589655	0%
	Set 2	4325681			2532	37556211	0%
	Set 3	4321560			2430	39952241	0%
	Set 4	4215358			2458	35697845	0%
FLR12_P6	Set 1	4362155			3232	39562321	0%
	Set 2	4254412			3125	39995654	0%
	Set 3	4325621			3238	38956621	0%
	Set 4	4698745			3312	40000455	0%

FLR14_P3	Set 1	6342652	3355	5785	9985	89654888	0%
	Set 2	6042325			10002	78528155	0%
	Set 3	6865231			9888	89222221	0%
	Set 4	6578624			9909	90002556	0%
FLR14_P6	Set 1	6021553			12563	109654821	0%
	Set 2	6103645			11354	112546357	0%
	Set 3	6999299			12965	119866217	0%
	Set 4	6156224			13658	109862478	0%
FLR16_P3	Set 1	8892500	4169	8076	19352	186358924	0%
	Set 2	8785200			20536	200028961	0%
	Set 3	8858900			18897	179872259	0%
	Set 4	8692800			18888	179885357	0%
FLR16_P6	Set 1	8536203			20285	230558418	0%
	Set 2	8876245			27982	232554457	0%
	Set 3	8635581			24852	245685212	0%
	Set 4	8766652			25668	248858834	0%
FLR18_P3	Set 1	9256220	4891	9226	18000	236853558	5.06%
	Set 2	9589335			18000	245382482	8.56%
	Set 3	9368952			18000	243258425	3.65%
	Set 4	9589255			18000	236585425	5.71%
FLR18_P6	Set 1	9986634			18000	280855554	10.50%
	Set 2	9286323			18000	286542255	12%
	Set 3	9998785			18000	296582358	11.05%
	Set 4	9185625			18000	281247828	11.69%
FLR20_P3	Set 1	10886235	5601	9853	18000	279832581	9.23%
	Set 2	10832592			18000	301015032	10.55%
	Set 3	10963252			18000	298658522	10.96%
	Set 4	10036541			18000	303562144	8.71%
FLR20_P6	Set 1	10963258			18000	329853852	15.04%
	Set 2	10956225			18000	332595474	16.11%
	Set 3	11364878			18000	328563585	14.56%
	Set 4	10859993			18000	331255521	15.89%

Table 4.6. Results of solving dynamic multi-floor OTLP with FFLP

Obviously, increasing the number of facilities increases the computational time. Thus, for instances where optimality was not guaranteed, we fixed the computational time and calculated the gap between the lower bound and the upper bound for each instance. Table 4.6 shows the best-obtained objective value in the planning horizon with FFLP approach, the number of variables and constraints, the computational time, the number of iterations and the gap for different sets of each instance size.

Figures 4.5 and 4.6 represent the retained solutions of the FFLP approach for instance with 14 facilities in the planning horizon of three periods (i.e. FLR14_P3) and for instance with 16 facilities in the planning horizon of six periods (i.e. FLR16 _P6) respectively.

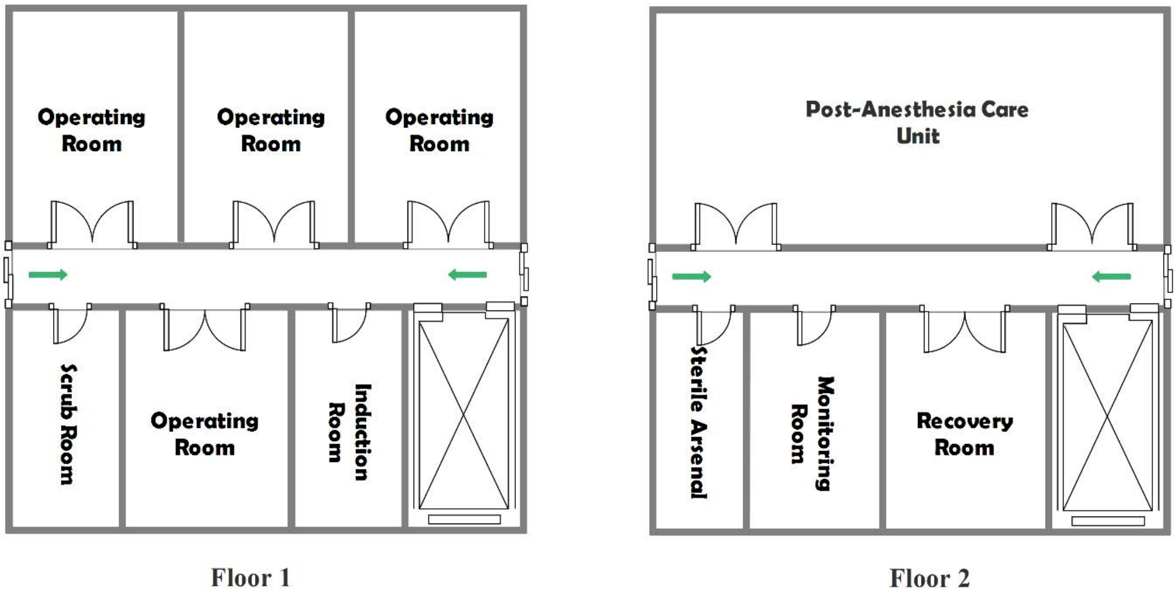


Figure 4.5. Robust layout of FFLP for FLR14_P3

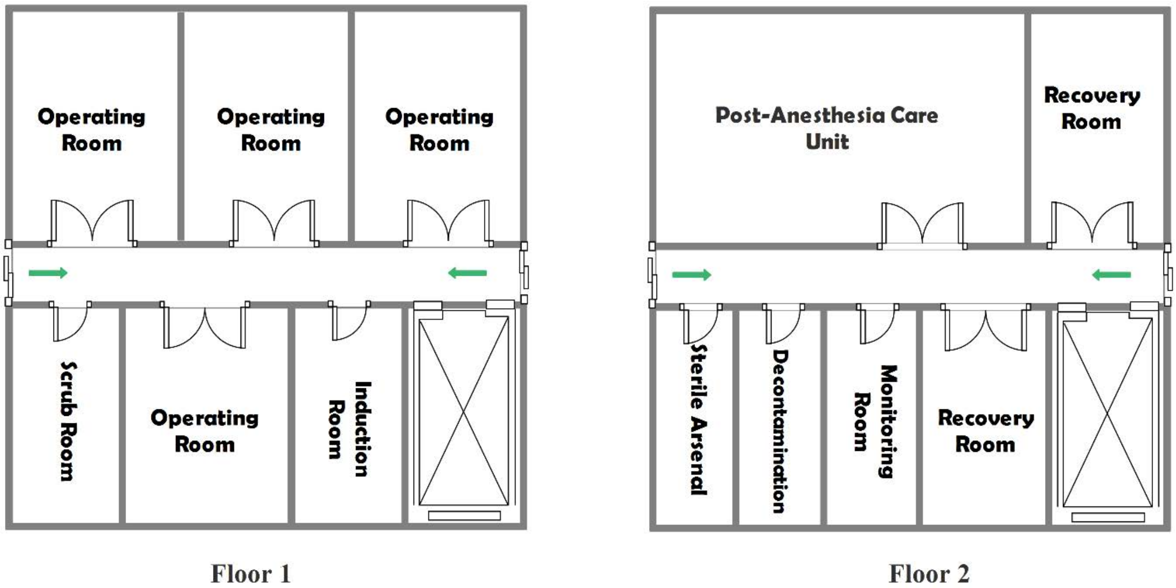


Figure 4.6. Robust layout of FFLP for FLR16_P3

ii) VFLP results

In VFLP approach, we consider that the operating rooms are the only facilities to exchange their position but not their floor. Our OR are composed of four operating rooms (e.g. general surgery, orthopedic surgery, Gynecology and Urology), assuming that when exchanging the position, all equipment related to the OR specialty are reallocated.

Table 4.7 presents the obtained results for instances of size twelve, fourteen and sixteen with a planning horizon of three periods (i.e. FLR12_P3, FLR14_P3 and FLR16_P3, respectively) with the number of reallocated facilities (i.e. column moves). Optimal solution was guaranteed for small instances (i.e. less than fourteen facilities), while near optimal solution was obtained for a problem of sixteen facilities.

#	Data	Objective	Variables	Constraint	Time (sec)	Number of Iteration	Gap	Moves
FLR12_P	Set 1	13595625	3500	5963	3201	163595242	0%	6
	Set 2	14956333			3325	178346900	0%	8
	Set 3	13354221			3286	168523685	0%	6
	Set 4	15824500			3220	183658524	0%	10
FLR14_P	Set 1	20435441	4963	6852	11025	292548921	0%	10
	Set 2	20985236			11391	286655482	0%	10
	Set 3	18963250			8921	253521542	0%	8
	Set 4	18963574			8702	248966357	0%	6
FLR16_P	Set 1	24850236	5842	8203	14583	350482120	5.25%	12
	Set 2	21566587			11054	325478245	4.02%	8
	Set 3	24896287			14803	345824447	5.36%	10
	Set 4	20153357			10378	300552258	3.66%	4

Table 4.7. Results of solving Dynamic multi-rows OTLP with VFLP

Figures 4.7 and 4.8 are representations of the optimal solutions obtained for multi-floor OTLP with the VFLP approach in a planning horizon of three periods.

4.1.1. Summary

This part describes two Mixed Integer Programing models for the Dynamic Multi-Floors OTLP and motivation of these formulations. A Fixed Facility Layout Problem and a Variable Facility Layout Problem were proposed to solve this formulation.

Optimal solution was obtained for problems with seventeen facilities and fourteen facilities for FFLP and VFLP respectively. The tested instances represent a case of small OT with a reduced number of facilities where the choice of facilities to consider was based on expert recommendations.

The next part of this chapter will introduce the dynamic formulation for the multi-rows OTLP and will present the different approaches to solve it.

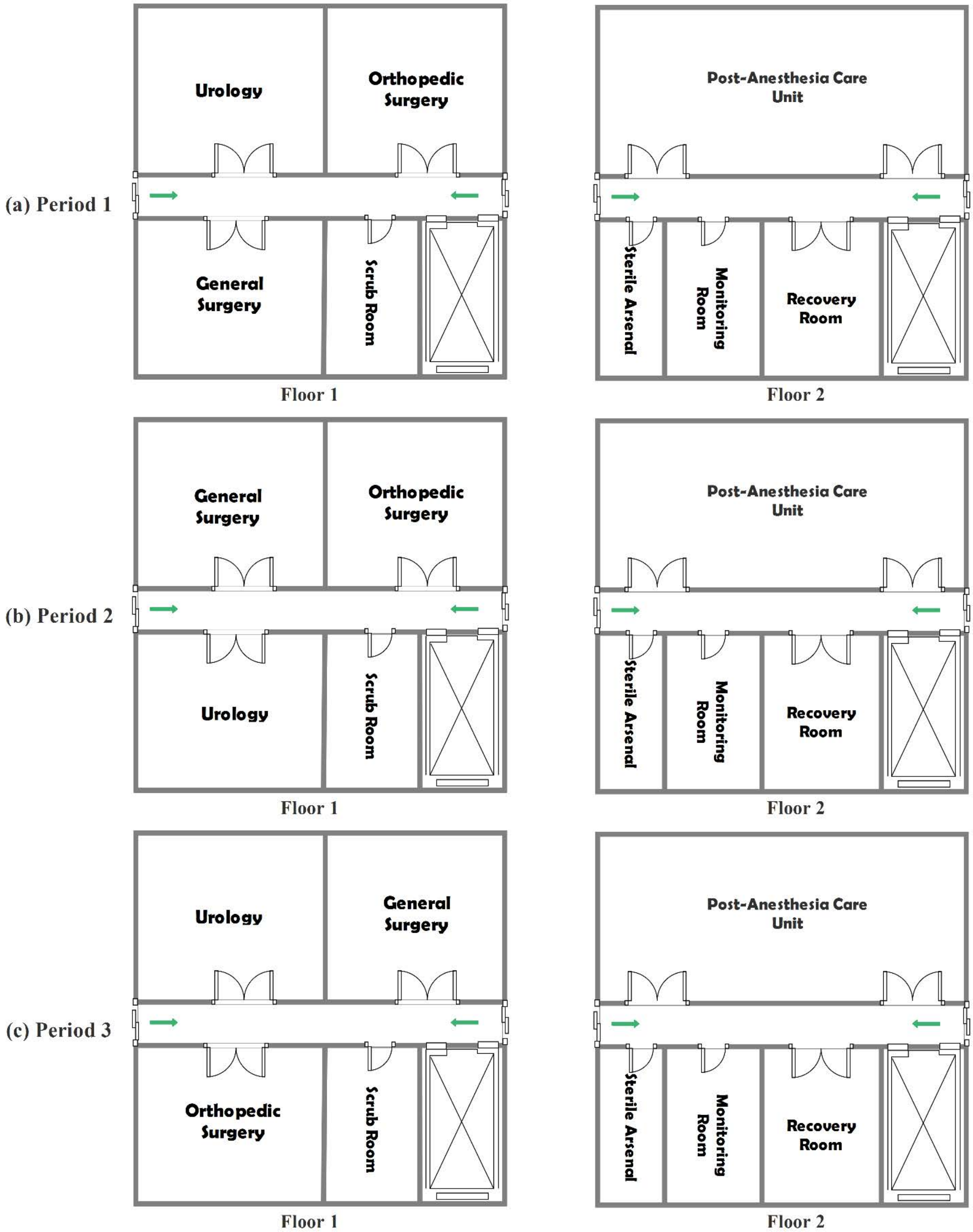


Figure 4.7. VFLP layout for FLR12_P3 for planning horizon of three periods

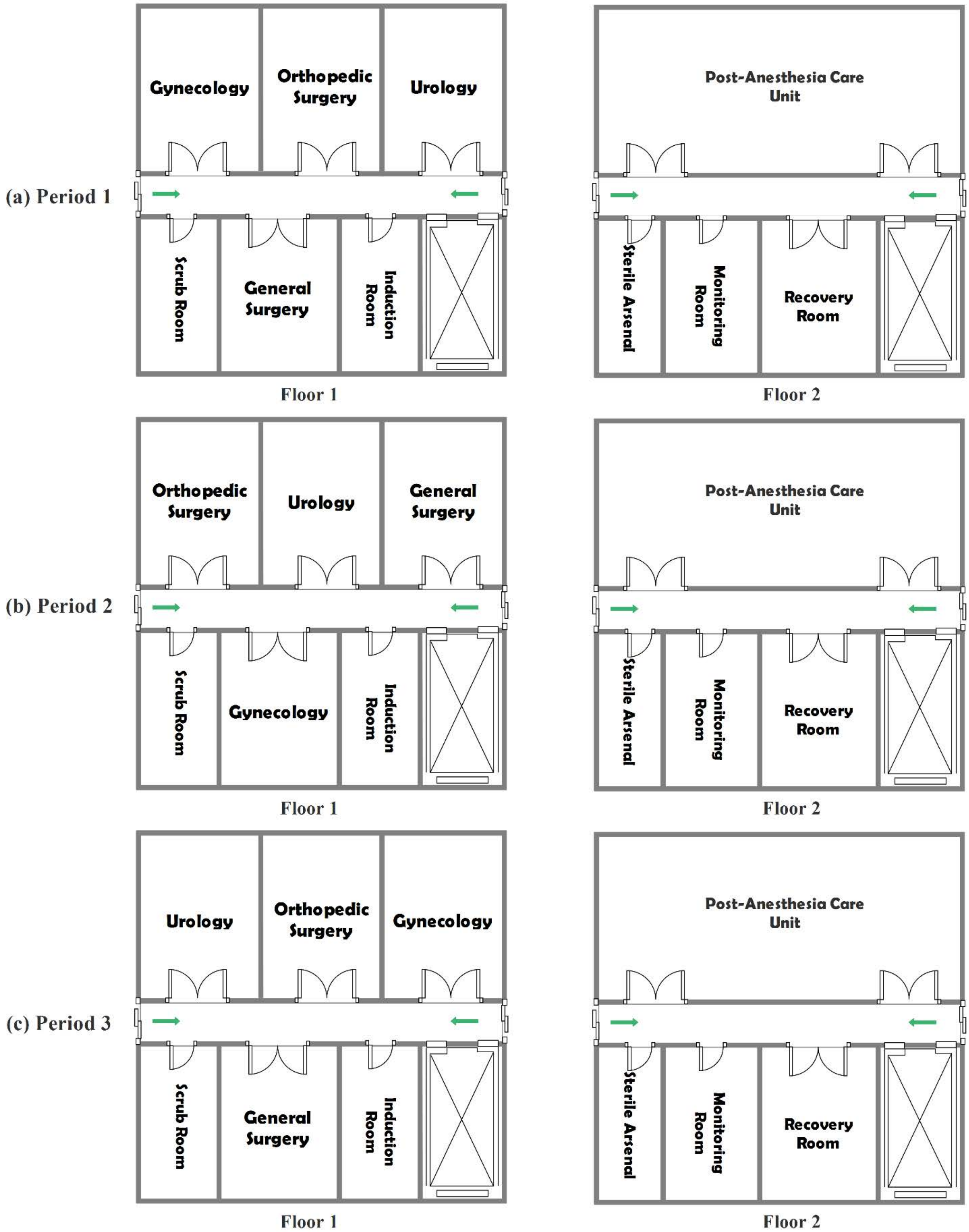


Figure 4.8. VFLP layout for FLR14_P3 for planning horizon of three periods

4.2. Mixed Integer Programming model for Multi-Rows OTLP

The multi-rows in static formulation proved to be sufficient with solving medium sized OTLP while reducing number of variables and constraints and decreasing the computational time. The dynamic multi-rows OT is also formulated as FFLP and VFLP, and keeps the same assumptions and constraints of the static formulation.

4.2.1. Mathematical Formulation

Sets

In addition to the sets of the static multi-row OTLP (cf. §3.5.1), we consider:

- P= $\{\tau_t$; t=1, 2,..., $p\}$ is the set of p periods in the planning horizon;
- NM_p is a set of a single element denoting the non-movable facilities.

Parameters

The new parameters in this formulation are:

F_{ijkt}:	The number of trips between facility a_i to facility a_j made by an entity type e_k at a period τ_t
Fm_{ijk}:	The average flow between facility a_i to facility a_j made by an entity type e_k for the various periods in the whole planning horizon
A_{ij}	Possibility of exchanging facilities affectation
C_{ij}	Relocation cost of facility a_i if shifted to a_j

Decision Variables

Ω_{it} { 1 if length α_i of facility a_i is parallel to x-axis at a period τ_t; 0 otherwise

μ_{ijt} { 1 if a_i and a_j are fully adjacent at a period τ_t; 0 otherwise

t_{ijt} { 1 if facility a_i and a_j are assigned to the same row at a period τ_t; 0 otherwise

v_{irt} { 1 if facility a_i is assigned to row r_r at a period τ_t; 0 otherwise

v_{ict} { 1 if facility a_i is adjacent to corridor θ_c at a period τ_t; 0 otherwise

o_{ijt} { 1 if facilities a_i and aj are adjacent to the same corridor θ_c at a period τ_t; 0 otherwise

z_{ijt}^{x}	{	1 if facility a_i is strictly to the right of facility a_j at a period τ_t 0 otherwise
x_{it} , y_{it}:		x and y coordinates of the centroid of facility a_i at a period τ_t
l_{it}:		x-length of facility a_i at a period τ_t
d_{it}:		y-length of facility a_i at a period τ_t
Xp_{ijt}:		x-distance between facility a_i and a_j in the same floor at a period τ_t
Yp_{ijt}:		y-distance between facility a_i and a_j in the same floor at a period τ_t
Xc_{it}:		x-distance between facility a_i and central corridor θ_c at a period τ_t
Y_{c1c2t}:		Vertical distance between corridor θ_{c2} and θ_{c2} at a period τ_t

i) FFLP Constraints

In this dynamic formulation, the FFLP proposes one robust layout for all periods of the planning horizon. Thus, we consider the average flow between each two facilities and calculate it using equation (4.4.1).

$$Fm_{ijk} = \frac{\sum_{t=1}^{P} F_{ijkt}}{P} \; \forall\, i, j \neq i, k \tag{4.4.1}$$

ii) FFLP objective function

In this model, we use a multi-objective MIP that aims to minimize the total traveling costs and to maximize the adjacency by considering the distance between facilities adjacent to the same corridor, and the distance between separated facilities (i.e. using central corridor to travel).

$$\boldsymbol{min\ F} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{4} \boldsymbol{Fm_{ijk}\,(Xp_{ij} + Yp_{ij} + Y_{c1c2})\varphi_{ijk} * \sigma_k} - \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{R_{ij}\mu_{ij}} \tag{4.4.2}$$

iii) VFLP constraints

Exposed constraints here are obtained after linearization:

$$l_{it} = \alpha_i \Omega_{it} + \beta_i (1 - \Omega_{it}) \qquad \forall i, t \tag{4.4.3}$$

$$d_{it} = \alpha_i + \beta_i - l_{it} \qquad \forall i, t \tag{4.4.4}$$

$$\sum_{r=1}^{R} v_{irt} = 1 \qquad \forall i, t \tag{4.4.5}$$

$$t_{ijt} = \left| v_{irt} + v_{jrt} - 1 \right| \; \forall\, i, j \neq i, r, t \tag{4.4.6}$$

$$t_{ijt} \geq v_{irt} + v_{jrt} - 1 \;\; \forall\, i, j \neq i, r, t \tag{4.4.7}$$

$$t_{ijt} \leq 1 - v_{irt} + v_{jrt} \;\; \forall\, i, j \neq i, r, t \tag{4.4.8}$$

$$t_{ijt} \leq 1 + v_{irt} - v_{jrt} \;\; \forall\, i, j \neq i, r, t \tag{4.4.9}$$

$$\sum_{c=2}^{C} v_{ict} = 1 \qquad \forall i\, , t \tag{4.4.10}$$

$$o_{ijt} \geq v_{ict} + v_{jct} - 1 \;\; \forall\, i, j \neq i, c \;\neq 1, t \tag{4.4.11}$$

$$o_{ijt} \leq 1 - v_{ict} + v_{jct} \;\; \forall\, i, j \neq i, c \;\neq 1, t \tag{4.4.12}$$

$$o_{ijt} \leq 1 + v_{ict} - v_{jct} \;\; \forall\, i, j \neq i, c \;\neq 1, t \tag{4.4.13}$$

$$x_{it} - x_{jt} + X_{max}\left(2 - z_{ijt}^{x} - t_{ijt}\right) \geq \frac{l_{it} + l_{jt}}{2} \quad \forall\, i, j \neq i, t \tag{4.4.14}$$

$$x_{jt} - x_{it} + X_{max}\left(2 - z_{ijt}^{x} - t_{ijt}\right) \geq \frac{l_{it} + l_{jt}}{2} \quad \forall\, i, j \neq i, t \tag{4.4.15}$$

$$z_{ijt}^{x} + z_{jit}^{x} \leq t_{ijt} \;\; \forall\, i, j \neq i, t \tag{4.4.16}$$

$$x_{it} + \frac{l_{it}}{2} \leq L_r \quad \forall\, i, t \tag{4.4.17}$$

$$y_{it} + \frac{d_{it}}{2} \leq D_r \quad \forall\, i, t \tag{4.4.18}$$

$$x_{it} - \frac{l_{it}}{2} \geq 0 \quad \forall\, i, t \tag{4.4.19}$$

$$y_{it} - \frac{d_{it}}{2} \geq 0 \quad \forall\, i, t \tag{4.4.20}$$

$$Yp_{ijt} \geq (y_{it} - y_c) + \left(y_c - y_{jt}\right) \forall\, i, j \neq i, c \;\neq 1, t \tag{4.4.21}$$

$$Yp_{ijt} \geq (y_{it} - y_c) + \left(y_{jt} - y_c\right) \;\forall\, i, j \neq i, c \;\neq 1, t \tag{4.4.22}$$

$$Yp_{ijt} \geq (y_c - y_{it}) + \left(y_c - y_{jt}\right) \;\forall\, i, j \neq i, c \;\neq 1, t \tag{4.4.23}$$

$$Yp_{ijt} \geq (y_c - y_{it}) + \left(y_{jt} - y_c\right) \forall\, i, j \neq i, c \;\neq 1, t \tag{4.4.24}$$

$$Xp_{ijt} \geq \left(x_{it} - x_{jt}\right) - (1 - t_{ijt})(X_{max} + Y_{max}) \;\; \forall\, i, j \neq i, t \tag{4.4.25}$$

$$Xp_{ijt} \geq \left(x_{jt} - x_{it}\right) - \left(1 - t_{ijt}\right)(X_{max} + Y_{max}) \;\forall\, i, j \neq i, t \tag{4.4.26}$$

$$Xc_{it} \geq (x_{it} - x_c) \;\; \forall\, i, c = 1, t \tag{4.4.27}$$

$$Xc_{it} \geq (x_c - x_{it}) \;\; \forall\, i\, , c = \; 1, t \tag{4.4.28}$$

$$Y_{c1c2t} \geq H_c \sum_{b=1}^{C} b\left(v_{ict} - v_{jct}\right) \;\forall\, i, j \neq i, c \neq 1, t \tag{4.4.29}$$

$$Y_{c1c2t} \geq H_c \sum_{b=1}^{C} b\left(v_{jct} - v_{ict}\right) \;\; \forall\, i, j \neq i, c \neq 1, t \tag{4.4.30}$$

$$Xp_{ijt} \geq Xc_{it} + Xc_{jt} - \left(t_{ijt} + v_{ict}\right)X_{max} \quad \forall\, i, j \neq i, c \neq 1, t \tag{4.4.31}$$

$$x_{it}{=}x_{it-1} \quad \forall\, i \in NM_p\, \forall\, t\ \in T \tag{4.4.32}$$

$$y_{ti}{=}y_{it-1} \quad \forall\, i \in NM_p\, \forall\, t\ \in T \tag{4.4.33}$$

$$v_{ict} = v_{ict-1}\, \forall i, r, t \tag{4.4.34}$$

In VFLP, some facilities exchange their position from a period to another. Thus, constraints (4.4.32) and (4.4.33) ensure that the non-movable facilities do not move during the planning horizon while the constraint (4.4.34) ensures that facilities remain adjacent to the same corridor all over the planning horizon.

iv) VFLP objective function

$$\begin{aligned} \boldsymbol{min\ F} = & \sum_{t=1}^{P}\sum_{i=1}^{N}\ \sum_{j=1}^{N}\sum_{k=1}^{4} \boldsymbol{F_{ijkt}(Xp_{ijt} + Yp_{ijt} + Y_{c1c2t})\varphi_{ijk} * \sigma_k} \\ & - \sum_{t=1}^{P}\sum_{i=1}^{N}\ \sum_{j=1}^{N} \boldsymbol{R_{ij}\mu_{ijt}} + \sum_{t=2}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{A_{ij}C_{ij}r_{tij}} \end{aligned} \tag{4.2.35}$$

The main objectives are minimizing the traveling costs and maximizing the adjacency rating for each period in the planning horizon; and the minimizing of rearrangement costs when reallocating movable facilities.

4.2.2. Experiments and results

The proposed formulations are validated using a set of generated data on OTFL optimization. We used ILOG CPLEX 12.5 software to solve the model using Windows 7 platform, Intel5® Core ™ i5-3380M CPU@ 2.90GHz and 8Go of RAM. For the computational study, we generate four sets of data for each instance size. We consider three instances' size, and a planning horizon of three periods, where a period refers to one month of flows.

Table 4.8 provides the notations of the test instances prefixed by ROW and gives for each instance the list of associated type of rooms where P3 and P6 refer to a planning horizon of 3 periods and 6 periods, respectively.

Instance's size	Description	Room list
ROW24_P3 **ROW24_P6**	An instance of twenty four facilities with a planning horizon of three and six periods	6 operating rooms, 3 induction rooms, 3 scrub room, 2 PACU, 1 decontamination, 1 sterile arsenal , 1 cleaning room, 2 bed storage, 2 monitoring and 3 recovery rooms
ROW28_P3 **ROW28_P6**	An instance of twenty eight facilities with a planning horizon of three and six periods	6 operating rooms, 3 induction rooms, 3 scrub room, 2 PACU, 1 decontamination, 1 sterile arsenal , 1 cleaning room, 2 bed storage, 3 monitoring, 4 recovery rooms and 2 changing rooms

ROW30_P3 **ROW30_P6**	An instance of thirty facilities with a planning horizon of three and six periods	6 operating rooms, 3 induction rooms, 3 scrub room, 2 PACU, 1 decontamination, 2 sterile arsenal , 1 cleaning room, 2 bed storage, 3 monitoring, 4 recovery rooms, 2 changing rooms and 1 radiology
ROW32_P3 **ROW32_P6**	An instance of thirty two facilities with a planning horizon of three and six periods	6 operating rooms, 3 induction rooms, 3 scrub room, 2 PACU, 1 decontamination, 2 sterile arsenal , 1 cleaning room, 2 bed storage, 3 monitoring, 4 recovery rooms, 2 changing rooms, 1 radiology, 1 secretariat and 1 reception office
ROW36_P3 **ROW36_P6**	An instance of thirty six facilities with a planning horizon of three and six periods	6 operating rooms, 3 induction rooms, 3 scrub room, 2 PACU, 1 decontamination, 1 sterile arsenal , 1 cleaning room, 2 bed storage, 2 monitoring, 4 recovery rooms, 2 changing rooms, 1 radiology, 1 blood test, 1 secretariat, 2 waiting rooms, 1 meeting room, 1 rest room and 1 reception office
ROW40_P3 **ROW40_P6**	An instance of forty facilities with a planning horizon of three and six periods	10 operating rooms, 5 induction rooms, 5 scrub room, 2 PACU, 1 decontamination, 2 sterile arsenal , 1 cleaning room, 2 bed storage, 3 monitoring, 4 recovery rooms, 2 changing room, 1 toilet, 1 radiology, 1 secretariat and 1 reception office

Table 4.8. Description of each multi-row instances

i) FFLP results

	Data	Objective	Variables	Constraint	Time (sec)	Number of Iteration	%Gap
ROW24_P3	Set 1	13235682	2305	8616	50	3562	0%
	Set 2	12526361			56	3661	0%
	Set 3	13254587			55	3425	0%
	Set 4	12895258			58	3924	0%
ROW24_P6	Set 1	12935482			70	4120	0%
	Set 2	13159547			75	4312	0%
	Set 3	13033255			74	4300	0%
	Set 4	12993588			76	4435	0%
ROW28_P3	Set 1	15368458	3137	11732	80	14369	0%
	Set 2	14935899			88	13596	0%
	Set 3	15023688			86	14953	0%
	Set 4	15196586			84	14302	0%
ROW28_P6	Set 1	15325987			102	16253	0%
	Set 2	14999866			110	16358	0%
	Set 3	14896587			108	16793	0%
	Set 4	15238150			116	16802	0%
ROW30_P3	Set 1	18956352	3601	13470	120	18635	0%
	Set 2	19034452			119	18963	0%
	Set 3	19126548			124	19856	0%
	Set 4	18998564			115	18425	0%

ROW30_P6	Set 1	19536584			138	21028	0%
	Set 2	19148621			145	21596	0%
	Set 3	19112573			149	22658	0%
	Set 4	19453216			141	21478	0%
ROW32_P3	Set 1	20956325	4097	14288	140	28485	0%
	Set 2	21035790			153	30256	0%
	Set 3	22153688			147	29436	0%
	Set 4	21036548			152	29946	0%
ROW32_P6	Set 1	20008535			173	35895	0%
	Set 2	21536548			169	35692	0%
	Set 3	22365480			172	35802	0%
	Set 4	21356458			180	36824	0%
ROW36_P3	Set 1	24635489	5185	19404	183	42035	0%
	Set 2	24987535			190	45266	0%
	Set 3	25003256			187	43658	0%
	Set 4	24977452			194	46235	0%
ROW36_P6	Set 1	25198788			210	49635	0%
	Set 2	25730256			223	50365	0%
	Set 3	25178532			219	49935	0%
	Set 4	25232154			225	51003	0%
ROW40_P3	Set 1	30125968	6401	23960	220	57022	0%
	Set 2	31256972			229	57235	0%
	Set 3	31459875			221	57651	0%
	Set 4	30989689			236	58596	0%
ROW40_P6	Set 1	31023658			257	60258	0%
	Set 2	32056981			244	59364	0%
	Set 3	31903535			261	61235	0%
	Set 4	32036900			249	59935	0%

Table 4.9. Results of solving dynamic multi-row OTLP with FFLP

Multi-rows formulation shows its performance also for dynamic formulation. Medium sized problems are solved to the optimum in a reasonable time for a planning horizon of three and six periods.

Examples of resulting layouts in Table 4.9 are presented in Figure 4.9 and Figure 4.10 for ROW24_P6 and ROW36_P3, respectively. In Figure 4.9, facilities in row 1 and row 2 are adjacent to the same corridor and traveling between each pair of facilities in these rows occurs through this corridor. On the other hand, when travel occurs between a facility in row 1 or row 2 to a facility in row 3 or row 4, the use of central corridor is unavoidable.

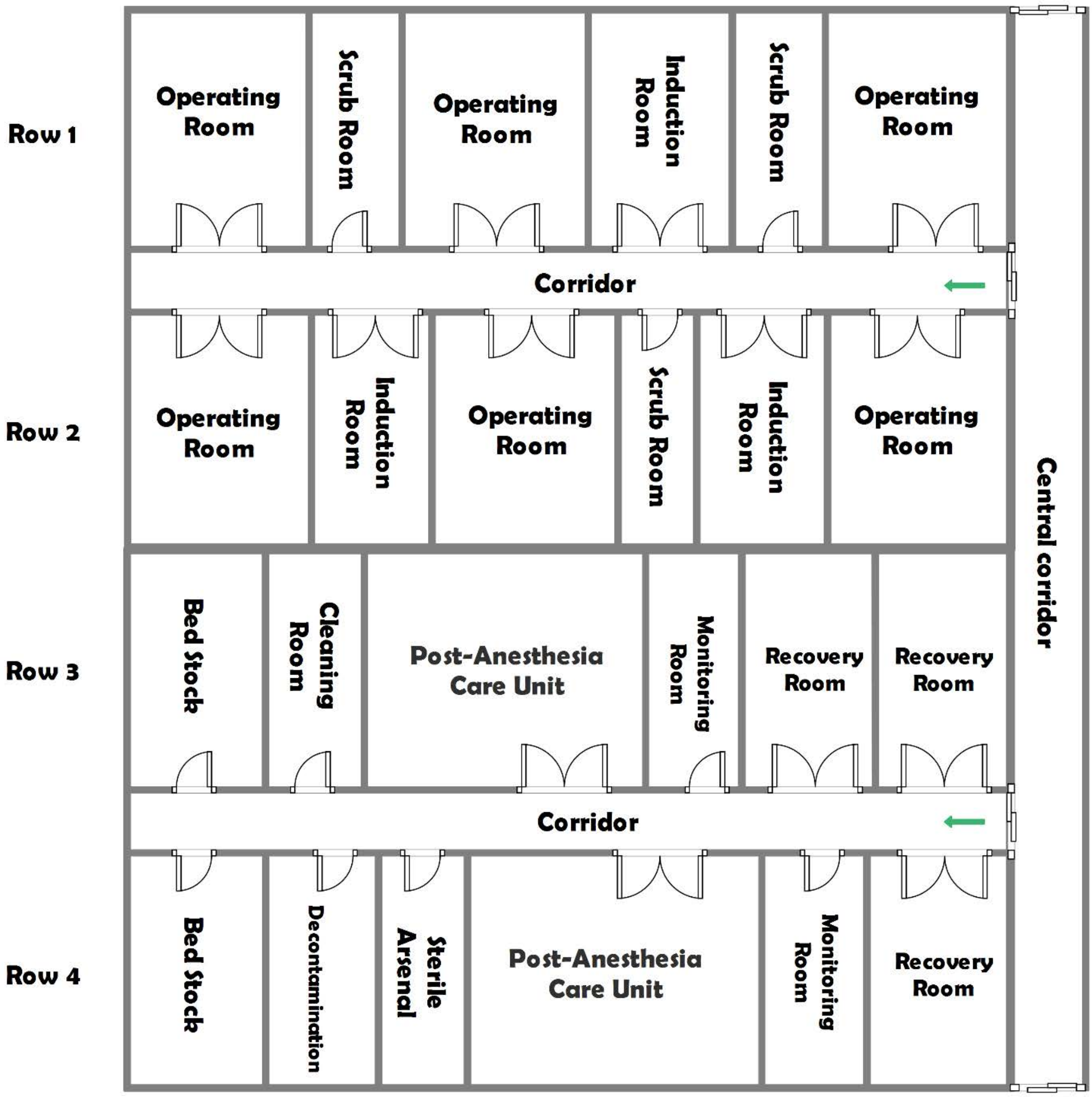


Figure 4.9. Robust layout of FFLP for ROW24

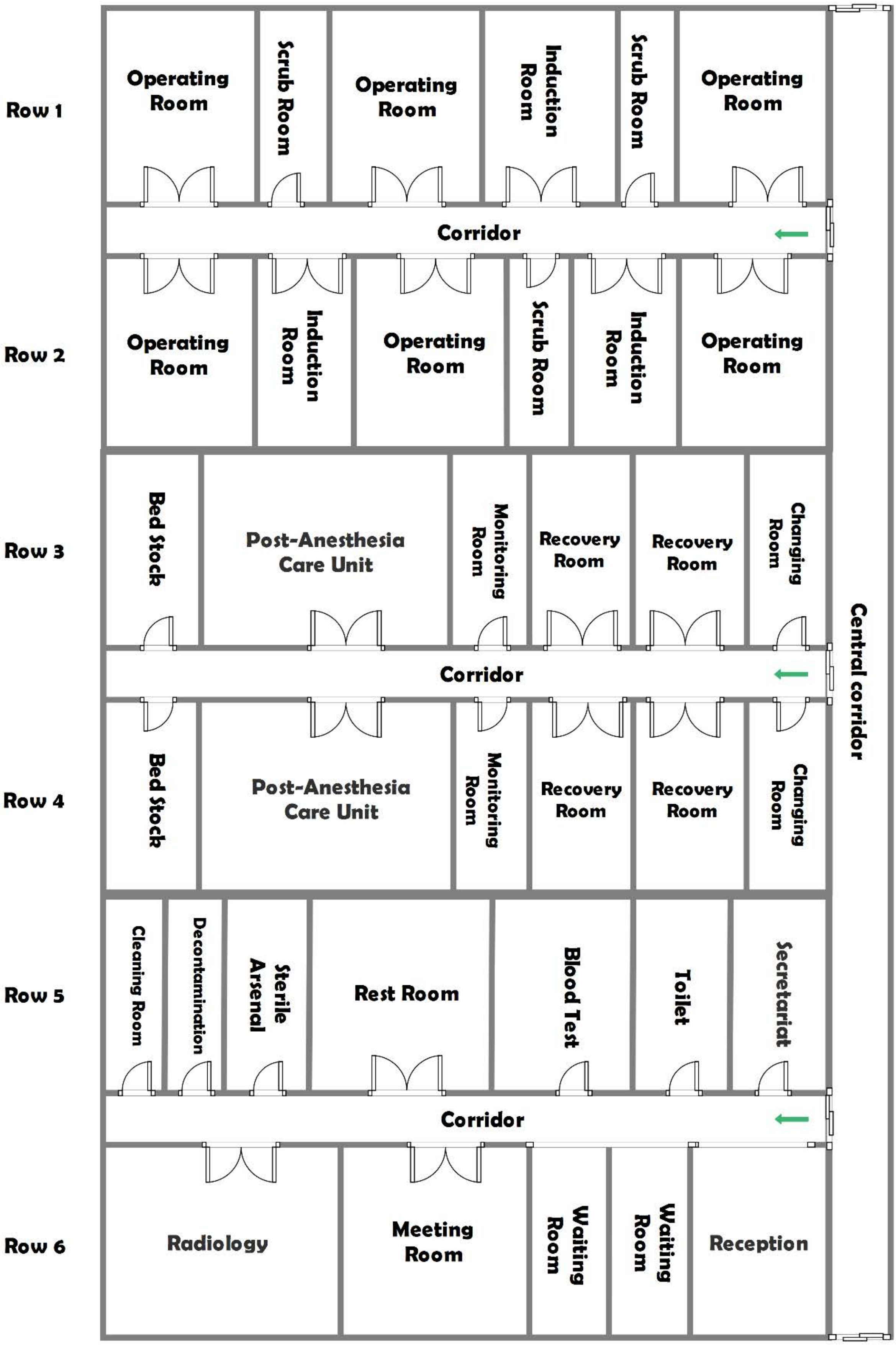


Figure 4.10. Robust layout of FFLP for ROW36

ii) VFLP results

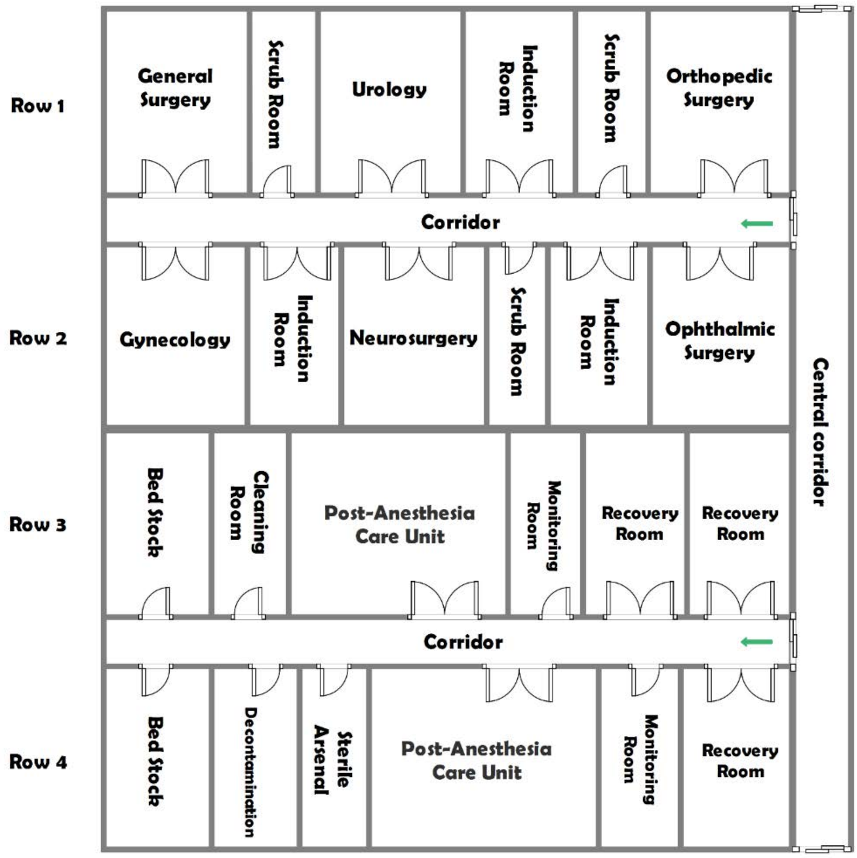
#	Data	Objective	Variables	Constraint	Time (sec)	Number of Iteration	Gap	Moves
ROW24_P3	Set 1	13695220	3596	12592	486	125886	0%	4
	Set 2	14895620			596	115969	0%	8
	Set 3	14325842			503	123265	0%	6
	Set 4	14955124			593	119862	0%	10
ROW28_P3	Set 1	1536952	3952	15030	789	156352	0%	8
	Set 2	1503509			900	163215	0%	6
	Set 3	1596325			865	160325	0%	10
	Set 4	1586384			835	159233	0%	10
ROW30_P3	Set 1	2136985	4562	18369	11235	19856	0%	8
	Set 2	2238942			12965	20136	0%	10
	Set 3	2153358			12358	21358	0%	8
	Set 4	2359944			11954	19230	0%	12
ROW32_P3	Set 1	2698272	5002	23028	17985	22985	0%	6
	Set 2	2654828			18032	23358	0%	8
	Set 3	2865823			18214	24828	0%	12
	Set 4	2789526			17802	22097	0%	10
ROW36_P3	Set 1	3025582	5705	28635	22582	28652	0%	8
	Set 2	2985637			21358	27452	0%	6
	Set 3	3152882			22968	29032	0%	10
	Set 4	3015982			23001	30128	0%	8

Table 4.10. Results of solving dynamic multi-row OTLP

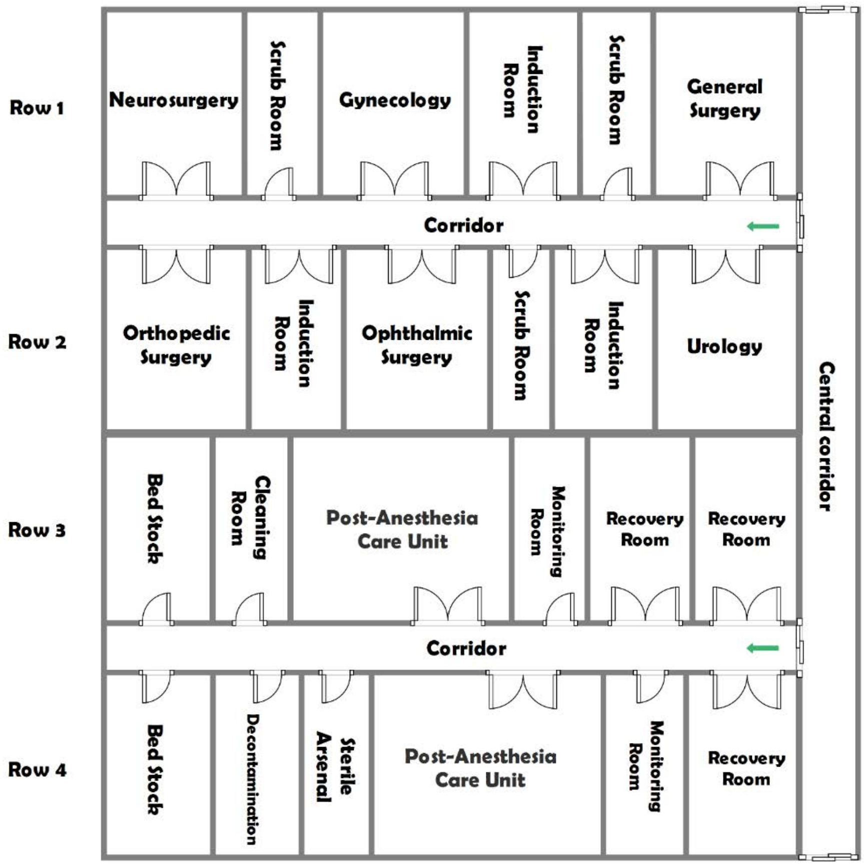
Table 4.10 presents results obtained while solving VFLP for multi-row OTLP. This formulation proved to be efficient even for solving the complex VFLP approach for which previous formulations (i.e. multi-sections and multi-floors) were not able to exceed sixteen facilities.

Figure 4.11 presents the layout evolution in each period of the planning horizon for a problem of twenty-four facilities. In this example, we consider six operating rooms' specialties (e.g. OR1= general surgery, OR2= orthopedic surgery, OR3= gynecology, OR4= urology, OR5= ophthalmic surgery and OR6= neurosurgery).

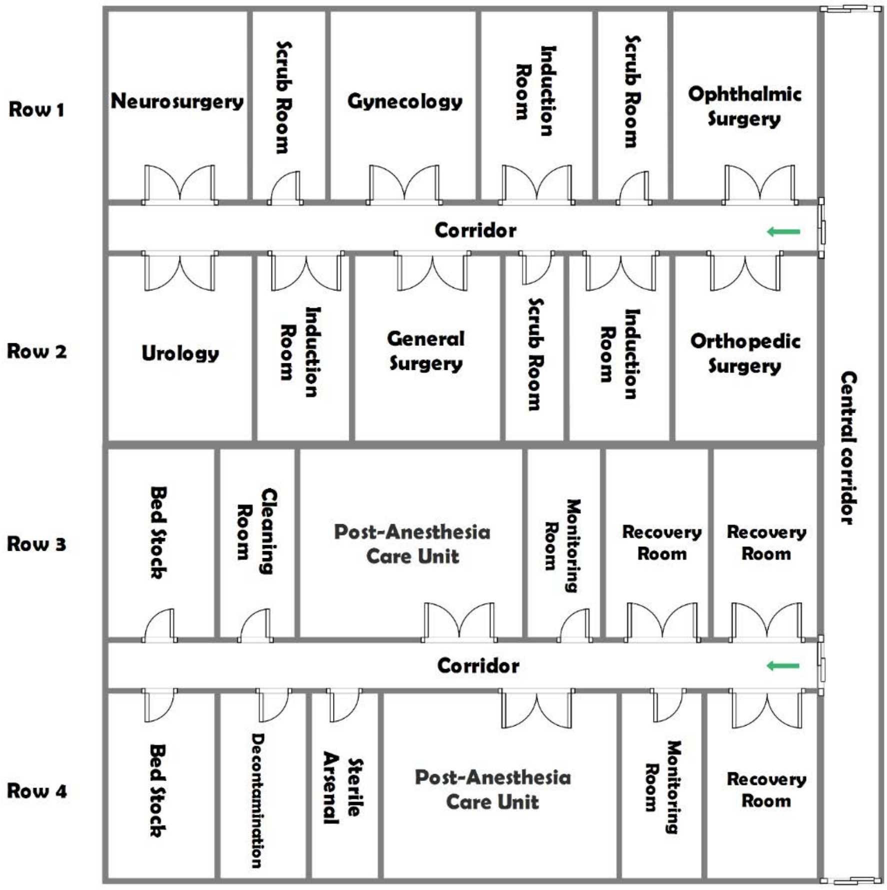
As assumed before, operating rooms are the only facilities to exchange their locations in order to follow the various surgery demands in different periods.



(a) Period 1



(b) Period 2



(c) Period 3

Figure 4.11. VFLP layout for ROW24_P3 for planning horizon of three periods

4.2.3. Summary

This part describes two Mixed Integer Programing models for the Dynamic Multi-Rows OTLP to deal with two different formulations. The first formulation aims to find in one single decision a robust OT layout to all periods of the planning horizon and which cannot be modified later. The second formulation aims to find a first layout for the first period, and then exchange some facility locations in the next periods according to surgical demand.

Optimal solution was guaranteed for medium sized problem for both FFLP and VFLP formulations with problems of fourteen and thirty-six facilities respectively. The obtained results proved the efficiency of the proposed model and gave inspiration to improve this model with other assumptions and considerations.

4.3. Conclusion

This chapter presents three MIP formulations to solve Dynamic OTLP using two different approaches. The FFLP aims to find in only one single decision a robust OT layout for the whole planning horizon. In contrast, the VFLP seeks to generate an OT layout for each period in the planning horizon. In this chapter, we tested the performance of the proposed formulations to solve Dynamic OTLP under several assumptions and constraints for different problem sizes and different planning horizon segmentations.

Due to unavailable realistic values of different actors flows, their costs, building and rearrangement costs, we generate a set of data for different instance size based on real life observations in some hospitals. The computational results were optimal with FFLP in a reasonable time for the three MIP formulations, while the designed layouts were satisfying international standards in term of safety and hygiene. For VFLP we did not attain optimality for bigger instance sizes in multi-section nor multi-floor OTLP, but final OT layouts show that even for the remaining gap, we obtain the same layout in some cases. The FFLP approach gave good performances that were equivalent to the static formulation results, while VFLP was not so far from the static formulation performance due to the complexity of this approach in terms of variables and constraints number.

This is a crucial advance as usually OT layouts are planned manually without taking into account the optimization aspects. The developed models can be used as a decision support tool to planners for optimal OT layout design since the considered assumptions and the obtained results were approved by a senior manager of OT in Roanne’s hospital.

Looking for optimality lead us to perform our models in order to reduce computational time and use bigger problem sizes. The next section will present an approximate method, which try to find near optimal solutions for large-sized problems in reasonable time.

4.4. Résumé

Ce chapitre présente trois formulations MIP pour résoudre l’OTLP dynamique en utilisant deux approches différentes. Le FFLP (disposition avec installations fixes) consiste à trouver en une seule décision, unique, un agencement robuste du bloc opératoire pour tout l'horizon de planification. En revanche, le VFLP (disposition avec installations amovible) cherche à générer un agencement du bloc opératoire pour chaque période de l'horizon de planification. Dans ce chapitre, nous avons testé les performances des formulations proposées pour résoudre l’OTLP dynamique sous plusieurs hypothèses et contraintes pour des problèmes de tailles différentes et des segmentations différentes de l'horizon de planification.

En raison de non disponibilité des valeurs réalistes des flux de différents acteurs, leurs coûts, des coûts de construction et de réarrangement, nous générons un ensemble de données pour des instances de tailles différentes fondées sur des observations réelles dans certains hôpitaux. Les résultats de calculs étaient optimaux avec le FFLP dans un délai raisonnable pour les trois formulations MIP, tandis que les dispositions conçues étaient satisfaisantes aux normes internationales en termes de sécurité et d'hygiène. Pour VFLP nous n'avons pas atteint l'optimalité pour les instances de grandes tailles dans le cas de l'OTLP en multi-section et en multi-étage, mais les dispositions finales des blocs opératoires montrent que, même pour l'écart restant, on obtient la même disposition dans certains cas du FFLP. L'approche du FFLP a donné de bonnes performances qui étaient équivalentes aux résultats de formulation statique, tandis que le VFLP n'était pas si loin de la performance de la formulation statique due à la complexité de cette approche en termes du nombre de variables et de contraintes.

Ceci est une avancée cruciale puisque la conception habituellement des blocs opératoires est planifiée manuellement sans prendre en compte les aspects d'optimisation. Les modèles développés peuvent être utilisés comme un outil d'aide à la décision pour les planificateurs pour optimiser la conception des blocs opératoires puisque les hypothèses considérées et les résultats obtenus ont été approuvés par un cadre supérieur du bloc opératoire à l'hôpital de Roanne.

La recherche de l'optimalité nous amène à améliorer nos modèles afin de réduire les temps de calculs et de traiter des problèmes de plus grandes tailles. La section suivante présentera une méthode approximative, qui visera à trouver des solutions quasi-optimale pour des problèmes de grandes dimensions dans un délai raisonnable.

Chapter V. Multi-Agent System for Operating Theater Layout Problem

5.1. Introduction

In order to solve large-sized FLPs, exact methods are inefficient and still incapable of offering optimal solutions. These limitations drive researchers to use heuristics or metaheuristics to find near optimal or at least satisfying solutions. However, provided solutions are approximate and not necessarily optimal, this encourages us to explore a new approach providing an exact and optimal solution: Multi-Agent Systems.

Multi-agent systems (MAS) have been developed in the context of distributed artificial intelligence and consist of a set of distributed cooperating agents each of which acts autonomously. They provide a novel approach to complex problems in a distributed manner where decisions are based on information processing from various sources of diverse nature (Woolridge & Jennings [1995]).

MAS raised several research issues about their conception, implementation and validation within a real case study. It includes the implementation of the following techniques and mechanisms:

- coordination strategies allowing groups of agents to solve problems effectively;
- protocols through which agents can communicate and reason about their communications;
- negotiation mechanisms allowing a set of agents to be conducted in an acceptable overall condition;
- techniques to detect and resolve eventual conflicts

- mechanisms allowing agents to maintain their independence while contributing to the overall functioning of the system.

In the reminder of this chapter, we will provide the second section to MAS, to introduce the concepts of agents, interactions and communication protocols between them. The third and fourth sections are devoted to the developed MAS for static and dynamic OTLP, respectively. Finally, a conclusion and perspective improvement will be exposed in the last section.

5.2. Multi-Agent Systems

The notion of MAS gives the idea of a system made out of numerous agents. The concept of agent remains the pivot of this field and the interpretation of this term must be the first step in the exploration of the multi-agent universe. The way agents interact between them will introduce how MAS are built.

5.2.1. Definitions

i) Agent

The definition of an agent varies greatly depending on the used field or system. An agent is a vague concept that can designate a physical entity (i.e. a computer, a processor, a human, etc.) or a formal entity (i.e. a processor, a program, etc.) according to the application domain.

Ferber (1995) defines an agent as “a physical or virtual entity:

- which is capable of acting in an environment,
- which can communicate directly with other agents,
- which is driven by a set of trends (as individual objectives or satisfaction function it seeks to optimize),
- which has its own resources (the sum of knowledge and means that are available to agents in a way they can act in their environment),
- which is able to perceive (but to a limited extent) its environment,
- which has only a partial representation of this environment (and possibly none),
- which has expertise and provides services,
- which can eventually reproduce itself,
- whose behavior tends to satisfy its objectives, taking into account the resources and skills available to it, and according to its perception, its representations and communications it receives.”

Wooldridge & Jennings (1995) defines the term agent as “a hardware system or (more often) software, which has the following characteristics:

- Autonomy: the agent acts without human intervention or other stakeholders, and a certain control over its actions and internal states;
- Social capabilities: the agent interacts with other agents using an agent communication language;
- Reactivity: the agent perceives its environment (which can be a physical world, a user via a graphical interface, a set of other agents, Internet, or all of these combined), and respond opportunistically to changes therein occur;
- Pro-activity: the agent does not simply to stimuli in its environment; it is also able to demonstrate behaviors headed through goals by taking initiatives."

Multi-Agent Systems

Ferber (1995) defines the MAS as "a system composed of the following elements (see Figure 5.1):

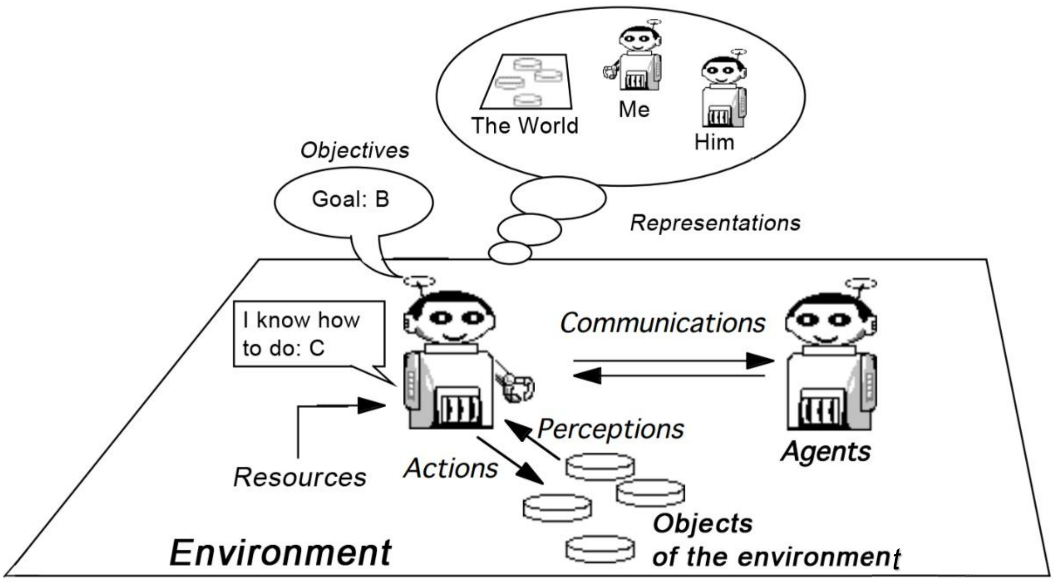


Figure 5.1. Pictorial representation of an agent interacting with its environment and other agents (Ferber [1995])

- An environment E, which is a space generally having a metric and representing the world in which agents evolve;
- A set of objects O. These objects are located, i.e. for each object, it is possible at any time to associate a position in E. These objects are passive, i.e. they can be viewed, created, modified and destroyed by the agents;
- A set of agents A, which are special objects representing the operating entities of the system;
- A set of relationships R uniting objects (and thus agents) to each other;
- A set of operations Op, allowing agents A to perceive, produce, consume, transform and manipulate objects O;

- Operators to represent the application of these operations and the reaction of the world to changes' attempt, which we call the laws of the universe."

Jennings et al. (1998) define the MAS as "*a loosely coupled network of agents that work together to solve problems that are beyond the capabilities or knowledge of individual agents.*"

ii) Cognitive and reactive agents

In MAS, there are two principal classes of agents according to their reasoning abilities: cognitive agents and reactive agents. The reactive agents are the most basic without intelligence, without anticipation and without planning. They have a reduced capacity protocol and language communication to operate with a Perception/Action cycle so that they react rapidly to asynchronous events without using complex reasoning. The reactive agent does not have a complete representation of its environment, neither itself and does not have a history of its actions. A MAS composed only of reactive agents could contain a large number of agents. Their behavior is then simply dictated by their relationship to their environment without having a representation of other agents or their environment.

On the other hand, cognitive agents are inspired from human behavior and are composed of intelligent agents having a knowledge covering all the information and expertise needed to achieve their tasks and to manage their interactions with other agents and with their environment. Cognitive agents act according to a perception/decision/action cycle, thus each agent has the ability to reason in terms of its own goals, its knowledge about other agents and its perception of the environment. Cognitive agents have the ability to communicate with other agents in a communication mode similar to human conversations.

Within each of these two classes of agents, there are different architectures that can be chosen according to the application field and the agents' functionality desired by the designer.

5.2.2. Agent interaction

Ferber (1995) defines an interaction as a dynamic linking of two or more agents through a series of reciprocal actions, which is both the source of its power and the source of its problems. The interaction can be divided into three phases [Chaib-draa (1996)]:

1. reception of information or perception of a change,
2. reasoning about other agents,
3. broadcast messages or actions having an impact on the environment.

In MAS, we can distinguish three forms of interaction: interaction by communication (coordination), cooperation and negotiation.

i) Coordination

In resource-constrained environments, coordination results in individual behavior to serve its own interests while trying to satisfy the overall goal of the system. In Jennings et al. (1998), coordination is characterized by two closely related aspects: commitments and agreements. The commitments provide the necessary structure for foreseeable interactions so that agents can take into account inter-agent dependencies of future facilities, global constraints or resource use conflicts. As situations change, agents should assess whether the existing commitments still valid. The agreements provide means to control the commitments in changing circumstances.

The actions of agents should be coordinated for several reasons:

- dependencies between agent actions,
- insufficient individual agent competence, resources and information to achieve by itself the objectives of the complete system or its own purpose,
- redundancy avoidance in problem solving.

Coordination is interested in how the actions of the agents are organized in time and space to accomplish the goals. Ferber (1995) distinguishes four types of coordination:

- Coordination through synchronization: It is the most basic form in which actions are precisely described in their enchainment.
- Coordination through planning: This technique is based on splitting each action into two phases. The first is to create a set of action plans that precisely describe actions to take to reach a goal. The second is the execution of one of these plans.
- The reactive coordination: This technique considers that it is easier to implement coordination mechanisms based on reactive agents than planning all the facilities.
- Coordination through regulation: It consists of ordering rules of conduct that aim at eliminating potential conflicts between agents.

ii) Cooperation

In cooperation, agents work to the satisfaction of an individual or a common goal, with the objective to improve the agents' working mode. Cooperative agents can change their goals to meet the needs of the other agents to ensure better coordination between them.

Cooperation aims to decompose a problem and dispatch the resulting tasks to different agents while specifying the role of each agent, the available resources for its task, the objective to satisfy and the constraints that it must overcome [Mazouz (2001)]. This strategy aims to reduce the complexity of the problem and optimize resource utilization, but at the same time, this decomposition risks generating interactions between tasks and therefore conflicts between agents.

The decomposition can be spatial or functional. Thus, a number of criteria must be respected:

- avoid overloading of critical resources;
- assign tasks to agents with corresponding capacities;
- assign interdependent tasks to agents spatially or semantically close to each other to limit communication costs;
- reassign tasks to accomplish the most urgent.

iii) Negotiation

Negotiation can be seen as a conflict resolution mechanism. This resolution can be performed in a favorable or unfavorable context. In a favorable context, all parties are going to make the necessary concessions to reach an agreement. While in an unfavorable context, the main goal for at least one of the participants is to get an agreement that favors his interest even at the expense of others. Therefore, the negotiation can be seen as cooperation with a common goal of obtaining an agreement.

The negotiation protocol is a process in which two or more agents have to take a common decision while each of them is trying to reach its own objective. Thus, to achieve the negotiation process several features are to be considered:

- the used language;
- the followed negotiation protocol by each agent;
- the decision-making process each agent uses to determine its positions, its concessions and its criteria for agreement.

5.2.3. Platform for developing MAS

Multi-Agent platforms have been developed to enhance the success of the multi-agent technology. The multi-agent platforms allow developers to design and implement their applications without wasting time performing basic functions for creating agents and coding interaction protocols between them. Furthermore, multi-agent platforms facilitate the task of developer and eliminate the need to be familiar with different theoretical concepts of multi-agent systems.

The most popular MAS platforms are SWARM [Burkhart (1994)], MADKIT [Gutknecht & Ferber (2001)], Zeus [Azvine (2000)] and JADE [Bellifemine & al., (1999)].

i) SWARM

SWARM is a generic platform allowing the development of agent-based simulations and consisting of a software library developed with C and later with Java at the Institute Santa Fe (New Mexico) by C. Langton (artificial life). SWARM has a strong orientation toward the development of discrete event simulation based applications for complex systems.

ii) MADKIT

MadKit (multi-agent Development Kit) is a generic platform developed in Java by Olivier Gutknecht within LIRMN Laboratory at Montpellier University. MadKit is used to design and execute MAS [Gutknecht & Ferber (1997)] via an execution engine wherein each agent is built starting from a microkernel.

MadKit allows the construction of MAS with a large number of agents and the establishment of synchronization between these agents during operation, where each agent has a role and may belong to a group. It also proposes a graphical development environment that allows easy construction of applications.

iii) Zeus

Zeus is a generic platform for MAS development, with possibility of customization since it could be configured by adding new components. This platform has been conceived by British Telecom (Agent Research Program of BT Intelligent Research Laboratory) based on the work of FIPA (Foundation for Intelligent Physical Agents) with the objective to develop collaborative applications. It is presented as a set of classes implemented in the Java language categorized into three functional groups: a component library, a development tool and a visualization tool.

Zeus has been used to develop several real applications such as auctions and simulation of manufacturing computers.

iv) JADE

JADE (Java Agent development framework) is a multi-agent platform developed in Java by the TILAB (Telecom Italia Lab) laboratory to enable the development of MAS and applications conform to the FIPA standards [FIPA (2002)]. The graphical interface of JADE is presented in Figure 5.2.

A JADE platform is composed of agent containers distributed over the network wherein agents live. Containers provide the JADE run-time and all the services needed for hosting and executing agents. A special case of containers is called the main container, which is the host of all agent containers and the bootstrap point of a platform and to which all containers must be linked and registered.

JADE has three main modules required in the FIPA standards, which are activated each time the platform is started:

- DF (Director Facilitor): provides a service "Yellow Pages" to the platform;
- ACC (Agent Communication Channel): handles communication between agents;
- AMS (System Management Agent): oversees the registration of agents, their authentication, access and use of the system.

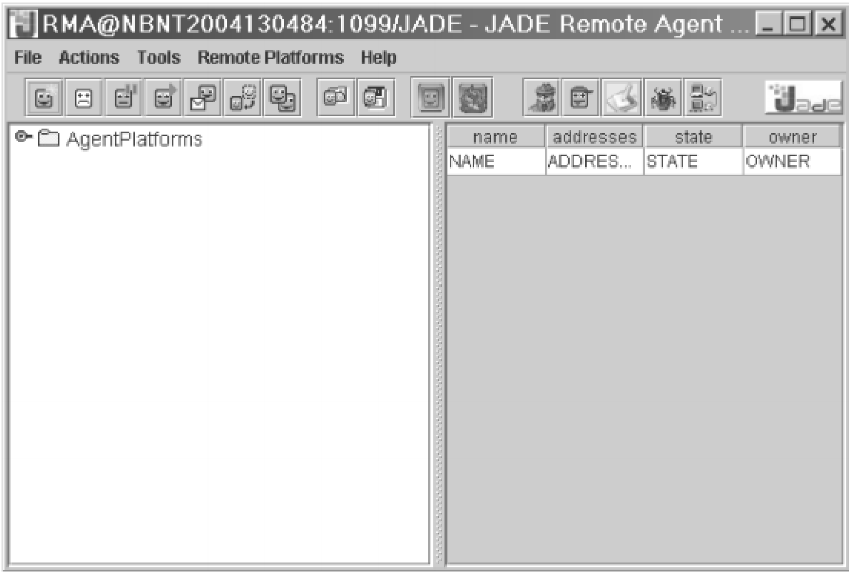


Figure 5.2. GUI of the JADE platform

After this brief presentation of some MAS development environments, JADE platform was more convincing to implement our distributed architecture. Some of the arguments and characteristics that led us to choose the JADE platform can be listed as follow:

- JADE is an agent platform meeting the specifications of FIPA,
- JADE provides several APIs for developers of Java agents,
- JADE is currently in full development and has all the possibilities to be extensible in the future,
- JADE runs on multiple operating systems, including Windows and Linux operating systems.
- JADE includes a ‘sniffer’, which is a graphical interface providing a log of all platform events and of all message exchanges between a set of specified agents (see Figure 5.3).

5.2.4. JADE exploitation

In this part, we will expose some of the Java code that implements the exploitation of JADE platform and different tools used in MAS:

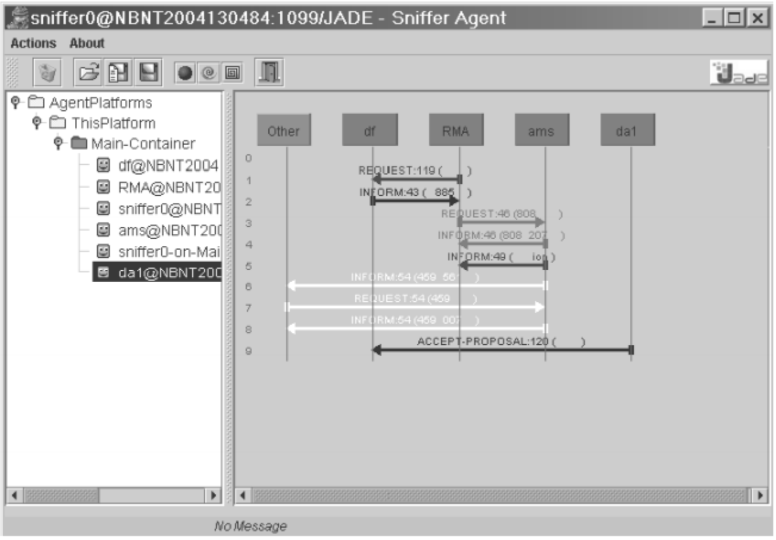


Figure 5.3. The Sniffer Agent GUI

Initialization and running of the JADE platform:

```
try {
      // Recuperation of the running Main Container in Jade platform:

      Runtime rt = Runtime.instance();
      // Creation of the default profile and the agent container:
      ProfileImpl p = new ProfileImpl(false);
      AgentContainer container =rt.createAgentContainer(p);
      // Agent controller for creating new agents
      AgentController Agent=null;
      // Creating the agent 'Agent_Name' (specify the agent's name) from
      the java class 'Agent_class' (specify the agent's class)
      Agent = container.createNewAgent("Agent_Name", "Agent_class", null);

      // Run the agent
      Agent.start();
      }
catch (Exception any) {any.printStackTrace();}
```

In order to publish a service an agent must create a proper description:

```
try { // Creation of the agent's description:
      DFAgentDescription dfd = new DFAgentDescription();
      dfd.setName(getAID());
      // Registering the Agent's description in DF (Directory Facilitator):
      DFService.register(this, dfd);
      }
catch (FIPAException e) {e.printStackTrace();}
```

When an agent terminates it is good practice to deregister published services:

```
protected void takeDown() {
      try {
            // Drop the agent from the DF
            DFService.deregister(this);
            }
      catch (FIPAException e) { e.printStackTrace();}
```

```
}
```

Once the main container is created, this fragment of code is used to generate slave processors:

```
// create N slave agents
try {
    for (int i = 0;  i < N;  i++) {
        // create a new agent
            // give a name to the created slave:
        String localName = "slave_"+i;
            // specify the slave java class
        AgentController guest = container.createNewAgent (localName,
         "Slave_Class", null);
        guest.start();
        // keep the guest's ID on a local list
        m_guestList.add( new AID(localName, AID.ISLOCALNAME));
    }
}
catch (Exception e) { e.printStackTrace();}
```

To exchange messages, we fill out the fields of an ACLMessage object and then calling the send() method of the Agent class:

```
// Send a message for all slaves from the coordinator agent
       ACLMessage m = new ACLMessage( ACLMessage.INFORM );
       m.setContentObject(Obj);
       m.addReceiver( Receiver );
       send( m );
```

Message reception must be continuously running (cyclic behaviors) and, at each execution of their action() method, a received messages must be checked and processed:

```
addBehaviour(new CyclicBehaviour(this) {
  public void action() {
      // Waiting for message from other agents
      ACLMessage msg = receive(MessageTemplate.
      MatchPerformative(ACLMessage.INFORM));
        if (msg != null) {
          try {
              // create an object from the received message
              Object[] obj=(Object[]) msg.getContentObject();
             // Treat the message according to agent’s tasks
             …
          }
          catch (UnreadableException e) { e.printStackTrace();}
          // delete the agent
                   doDelete();
             }
             else {
                 // While waiting for messages, block all other behavior of
                 this agent
                   block();
             }
      }
});
```

5.2.5. The use of MAS in the literature

Today, the interest of using and implementing of MAS no longer needs to be demonstrated. The research on complex agent-oriented systems has a wide variety of industrial, business, science and educational applications. The design of these systems requires adequate methodologies to cope with agent-specific characteristics, which result from agent-oriented domain analysis.

MAS are used in different domains. Nataraj Urs et al. (2010) presented a distributed MA framework for the VLSI layout design process. They divided the entire floor plan task into many subtasks, and affected each one to an agent to solve it. Finally, they make agents interact with each other to negotiate the final placement for the design process. Tarkesh et al. (2009) are the only ones to present an algorithm using a society of virtual intelligent agents to efficiently solve the FLP introducing thereby the concepts of emotion for the first time in operational research. They modeled each agent as a facility having proprieties such as money and emotions, which are adjusted during the agents' interactions, defined by market mechanism in which the richer agents can pay extra money to obtain better locations and are less interested in lower ones.

Cossentino et al. (2011) presented an MA simulation tool for decision making in automatic warehouse management. Their proposed MAS aims to optimize the suitable number of Automated Guided Vehicles used for unloading containers arrived to the warehouse. Nfaoui et al. (2008) developed an agent-based distributed architecture for collaborative decision-making processes within the global distribution supply chain for best management of the unexpected rush order for which the quantity of products cannot be delivered partially or completely from the available inventory.

Otherwise, in healthcare, Isern et al. (2010) present a review of the literature about applications of agents regrouping more than 15 works and classify each one according to the main goal of the systems. González-Vélez et al. (2009) presented an agent-based distributed decision support system for the diagnosis and prognosis of brain tumors developed by the HEALTHAGENTS project. Koutkias et al. (2005) presented an interactive MAS for the management of patients' chronic diseases with the objective to detect anomalous cases and informs personnel to enhance monitoring, surveillance, and educational services of a generic medical contact center.

Works using MAS are being increasingly conducted to discover more application fields to offer intelligent methods for solving complex problems in a distributed architecture.

5.2.6. Summary

This chapter allowed us to review briefly the MAS. We presented the key concepts that define this field, the methodologies, the MAS development platforms and a brief literature review of works using MAS in different domains. The choice of JADE platform to develop our MAS has been argued and some code fragments for agent deployment have been exposed.

The next sections present the designed MAS architectures and the proposed optimization formulation to solve the OTLP under static and dynamic environments, and for each one, expose the results of running the developed approach with several instances sizes.

5.3. Static Operating Theater Layout Problem

The use of MAS in FLP is practically non-existent. Tarkesh et al. (2009) are the only ones to present an algorithm using a society of virtual intelligent agents to efficiently solve the FLP introducing by this the concepts of emotion for the first time in operational research. They modeled each agent as a facility having proprieties such as money and emotions, which are adjusted during the agent's interactions, defined by market mechanism in which the richer agents can pay extra money to obtain better locations and are less interested in lower ones. To the best of our knowledge, it remains the only work dealing with FLP using MAS approach.

Our work comes to fill this void by proposing a first work dealing with OTLP using MAS. In our approach, we used cognitive and communicative agents. To communicate, they need a developed language to be able to exchange messages. However, a simple model of message is not enough to tolerate conversations between agents and to allow language acts to fulfill their meaning. For this, we introduced the notion of protocol to support such conversations. The negotiation strategy is proposed based on the Contract Net protocol implemented using the JADE platform that aims to simplify the development of MAS while providing a complete set of services and agents comply with FIPA specifications (FIPA, A. C. L. [2002]).

5.3.1. Formulation

The newly adopted approach is modeled in three levels of abstraction (see Figure 5.4); the first one is to define the dimensions and orientation of the plant layout or the initial construction site. The second level consists on using the MILP formulation to calculate the placement of each department in the initial plant layout. Finally, for each department, the third level is to solve the LP for the optimal facility locations and orientation.

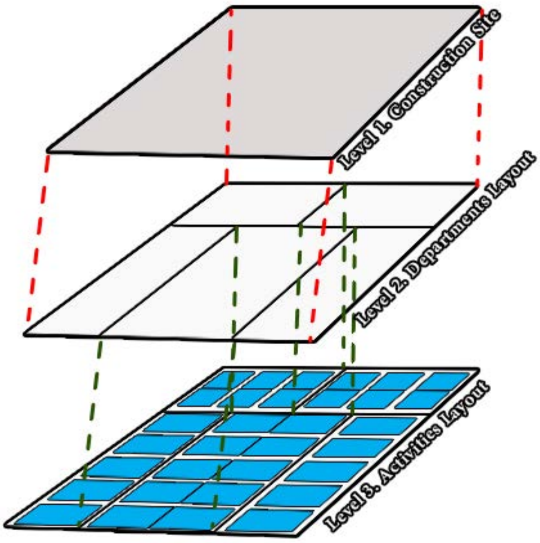


Figure 5.4. The three levels FLP approach

This three-step decision making approach can best be handled through an effective and efficient cooperation of multiple agents. To deal with this, we adopt a Master/Slave architecture where the Master Processor (MP) is the coordinator of the optimization process, while the Slave Processor is the worker executing tasks coming from the MP. The main task of an MP is to dispatch parameters and orders on generated Slave Processors (SPs) which solve the sub-problem, negotiate the best solution if received instances are the same and send the answer with objective value (see Figure 5.5).

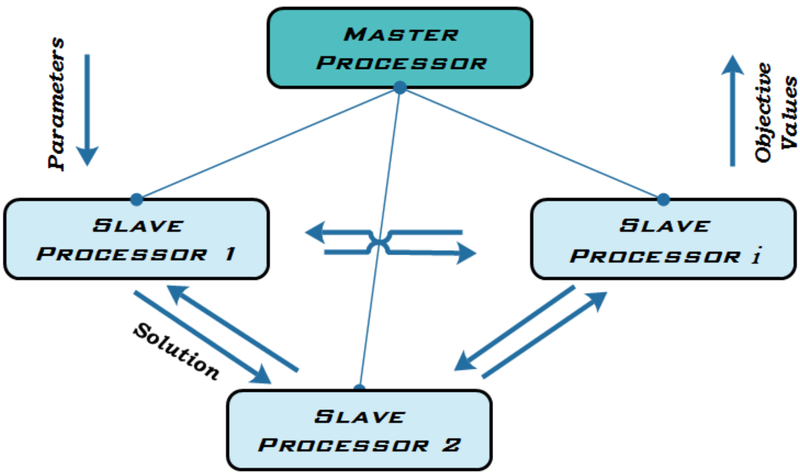


Figure 5.5. The Master/Slave architecture

The problem is modeled as a multi-department OTLP, where each department is solved as a multi-section problem in §3.3. The used algorithm is illustrated with a sequence diagram in Figure 5.6 and can be described as follow:

Step 1. Initialization

The initiator launches the mainContainer (JADE platform); setup the graphic Interface (GUI); initialize program parameters and register the MP.

Step 2. Solve Departments Layout

Once the MP registered, it divides the OT into sections according to their specialty and as imposed by hospitals managers. The MP starts solving the OTLP for the list of departments, and returns the location and the orientation of each one in the plant. These results will be parameters of the next step.

Step 3. Solve each Department Layout

[1] The MP creates and registers a SP for each department and uses the previous results to impose bounding constraint in solving sub-problems;

[2] Each SP first receives the instance of the sub-problem from the MP and starts solving MILP formulation using the equation (5.3.9) for the facilities and corridors in the department;

[3] Once a SP finishes solving its sub-problem, it sends the optimal layout with facilities' location and orientation to the other SPs:

> If instances are the same, each SP compare it solution with the received solution and consider the optimal solution.

[4] Once SPs finish negotiation, they send the retained solutions to the MP;

[5] The step stops when the MP collects all solutions.

Step 4. Drawing Solution

The result of SPs' collaboration is a final optimal OT layout. The MP sends this solution to the GUI that draws it and shows it to the decision makers to validate it.

Step 5. Agents' destruction

Finally, the MP informs the MainContainer about the end of process and destructs all working agents in the JADE platform.

This study on MAS allowed us to define how the capacity of cognitive agents can be used in DMS, namely their intelligence, cooperation and learning.

To make our MILP suitable for the three-level approach we have to add some parameters, variables and constraints related to department LP; to insure routing between them and to calculate the total traveling cost:

$$OrD_d(Xp_{ij}) + (1 - OrD_d)(Yp_{ij}) \geq OrD_d[(x_i - x_c) + (x_c - x_j)] + (1 - OrD_d)[(y_i - y_c) + (y_c - y_j)] \quad \forall\, i, c \text{ where } U_i = U_c, \forall d \tag{5.3.1}$$

$$OrD_d(Xp_{ij}) + (1 - OrD_d)(Yp_{ij}) \geq OrD_d[(x_i - x_c) + (x_j - x_c)] + (1 - OrD_d)[(y_i - y_c) + (y_j - y_c)] \quad \forall\, i, c \text{ where } U_i = U_c, \forall d \tag{5.3.2}$$

$$OrD_d(Xp_{ij}) + (1 - OrD_d)(Yp_{ij}) \geq OrD_d[(x_c - x_i) + (x_c - x_j)] + (1 - OrD_d)[(y_c - y_i) + (y_c - y_j)] \quad \forall\, i, c \text{ where } U_i = U_c, \forall d \tag{5.3.3}$$

$$OrD_d(Xp_{ij}) + (1 - OrD_d)(Yp_{ij}) \geq OrD_d[(x_c - x_i) + (x_j - x_c)] + (1 - OrD_d)[(y_c - y_i) + (x_j - x_c)] \quad \forall\, i, c \text{ where } U_i = U_c, \forall d \tag{5.3.4}$$

$$OrD_d(Yp_{ij}) + (1 - OrD_d)(Xp_{ij}) \geq OrD_d(y_i - y_j) + (1 - OrD_d)(x_i - x_j) \quad \forall\, i, j \neq i, d \tag{5.3.5}$$

$$OrD_d(Yp_{ij}) + (1 - OrD_d)(Xp_{ij}) \geq OrD_d(y_j - y_i) + (1 - OrD_d)(x_j - x_i) \quad \forall\, i, j \neq i, d \tag{5.3.6}$$

Where $\boldsymbol{OrD_d}$ refers to each department orientation which takes 1 if the length of department 'd' is parallel to y-axis (vertical orientation), 0 otherwise. We can obtain it by using:

$$W_d = \vartheta_d OrD_d + \theta_d(1 - OrD_d) \qquad \forall d \tag{5.3.7}$$

$$L_d = \vartheta_d + \theta_d - D_d \qquad \forall d \tag{5.3.8}$$

Where:

- $\boldsymbol{\vartheta_d}$: Length of department 'd'; (parameter)
- $\boldsymbol{\theta_d}$: Width of department 'd'; (parameter)
- $\boldsymbol{L_d}$: X-length of department 'd' depending on whether ϑ_d or θ_d is parallel on x-axis; (decision variable)
- $\boldsymbol{W_d}$: Y-length of department 'd' depending on whether ϑ_d or θ_d is parallel on y-axis; (decision variable)

The objective function in the second level will be formulated in (5.3.9) where F_{lhk}, D_{lh}, φ_{lhk}, R_{lh} and μ_{lh} are respectively inter-departments' traveling frequency, distance between, traveling difficulty, relationship and adjacency coefficient:

$$\boldsymbol{min\ FD = \rho_1 \sum_{l=1}^{D} \sum_{h=1}^{D} \sum_{k=1}^{4} F_{lhk} D_{lh} (\varphi_{lhk} * \sigma_k) - \rho_2 \sum_{l=1}^{D} \sum_{h=1}^{D} R_{lh} \mu_{lh}} \qquad (5.3.9)$$

5.3.2. Experiments and results

The proposed MILP is validated on OTLP using a set of generated data. We used ILOG CPLEX 12.5 software to solve the model using Windows 7 platform, Intel5® Core ™ i5-3380M CPU@ 2.90GHz and 8Go of RAM. For all instances, $\rho_1 = \rho_2 = 0.5$, $\sigma_{doctor} = 80$, $\sigma_{medical\ staff} = 60, \sigma_{non-medical\ staff} = 40\ and\ \sigma_{patient} = 20$ are used.

With MAS, we conserved the same instances and same sets of data used for the exact formulation in §3.3.2 and implemented the algorithm using JAVA language. We divide the program into three packages; the first contains all the required class to model the JADE agents, the second class models the different OT components (OT, departments, facilities, layout solution) while the third one is to design the GUI. The package Diagram showing this organization is modeled in Figure 5.7.

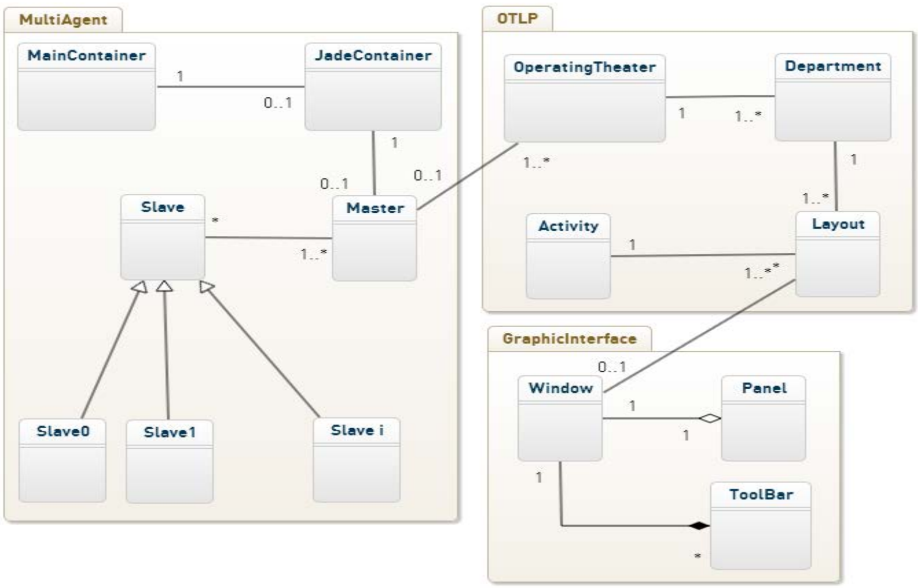


Figure 5.7. MAS package diagram

We test larger sized OTLP going until 12 departments with 162 facilities using 12 SP agents. The number of facilities being no longer a limitation while the sub-tasks are feasible MILP for each agent, we decide to stop at this number of facilities (162) to deal with realistic OT sizes. Compared to previous results obtained, this program gives an efficient solution for

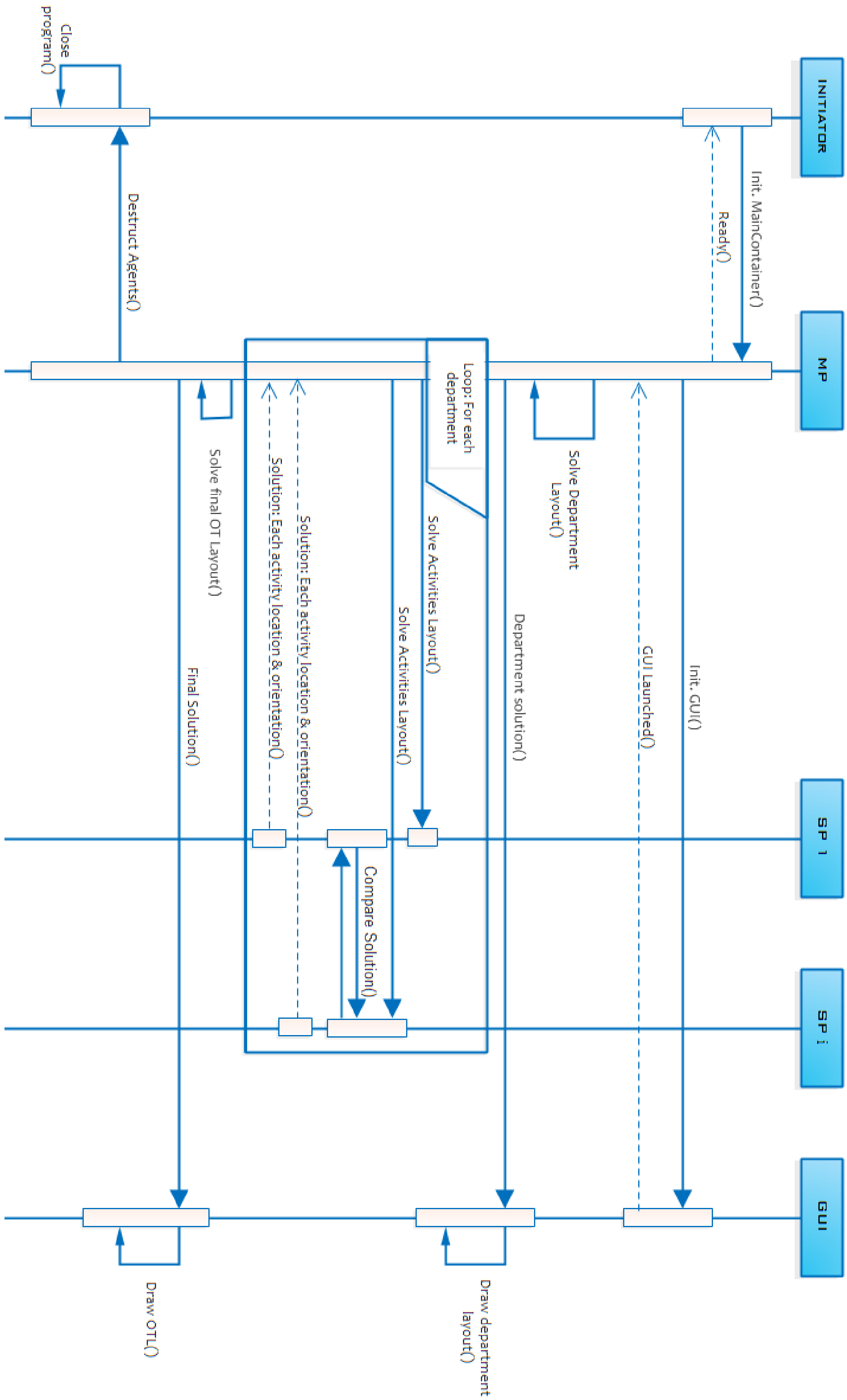


Figure 5.6. Sequence diagram of the developed approach

larger-sized problems in a shorter time. For example, a problem with sixteen facilities took

623sec with simple MIP, while MAS solved in the same duration problem with 108 facilities. This shows the performance of this MA approach in term of timeliness and capacity.

To be feasible, each agent's task consists of solving sub-problems with maximum size of sixteen facilities. Thus, when solving the multi-department problem, two agents will have to deal with different department sizes while respecting the maximum number of facilities in each department. Table 5.1 provides results of solving OTLP using MAS in terms of instance sizes, the number of agents used to solve it and the processing time for different running iteration.

We can also observe from table 5.1 the ability of agents to learn from their previous experiences. We repeated solving the same problem for several times for examining this agents' characteristics. Figure 5.8 shows that time processing regresses while repeating calculation for each problem size and it could have been more apparent if we had made more tests.

Number of agents	**Number of facilities**	**Time (sec)**		
		Iteration 1	***Iteration 2***	***Iteration 3***
2	27	194	188	163
4	54	389	372	332
6	81	517	506	489
8	108	672	669	657
10	135	927	908	888
12	162	1276	1137	1086

Table 5.1. Processing time for different instances sizes of the static formulation

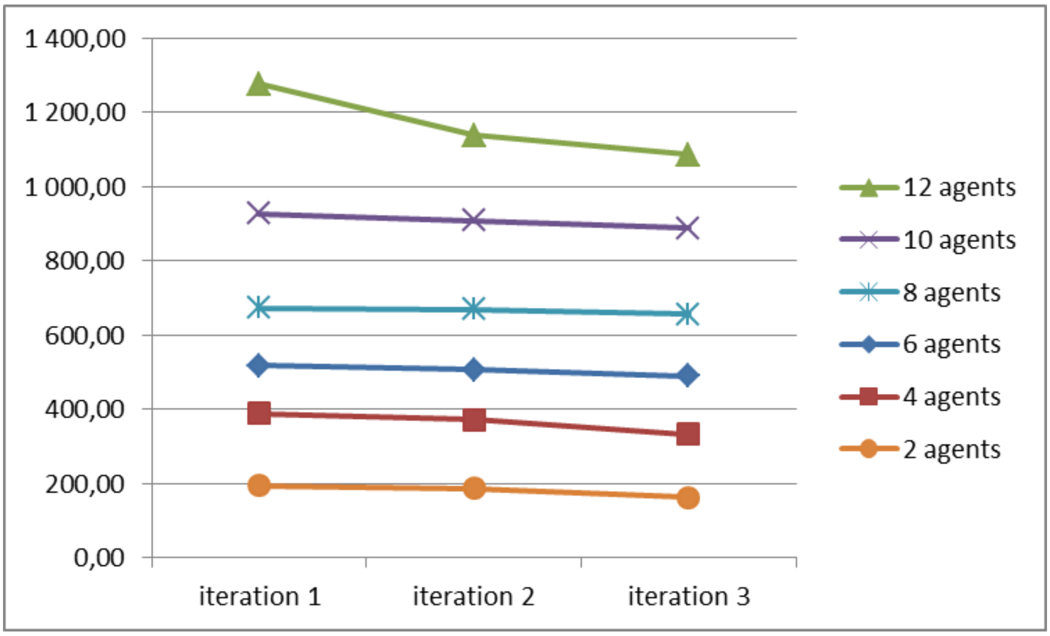


Figure 5.8. Processing time regression

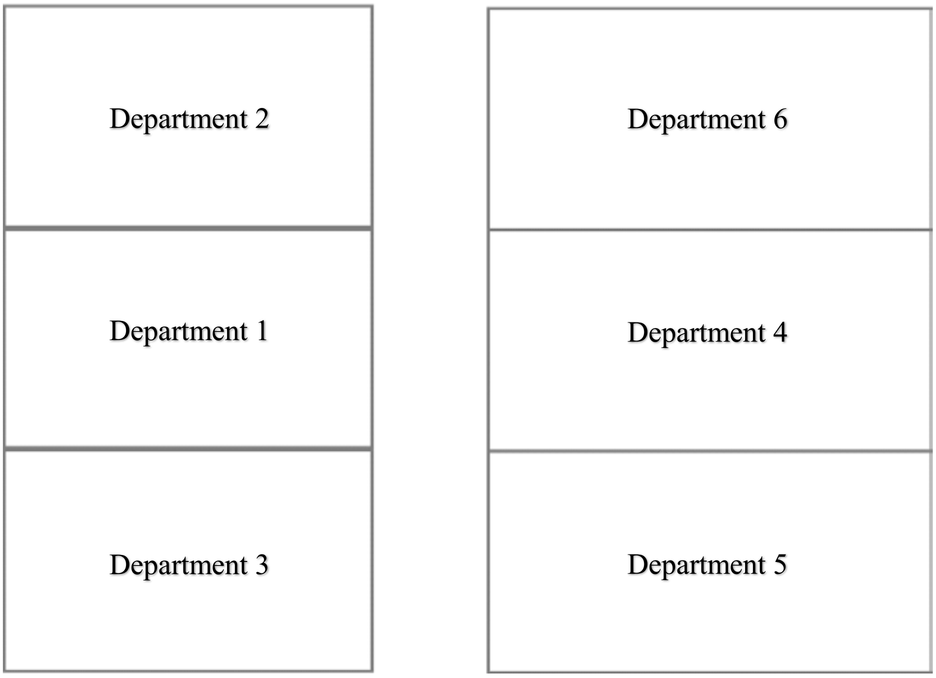


Figure 5.9. Departments Layout obtained in step 2

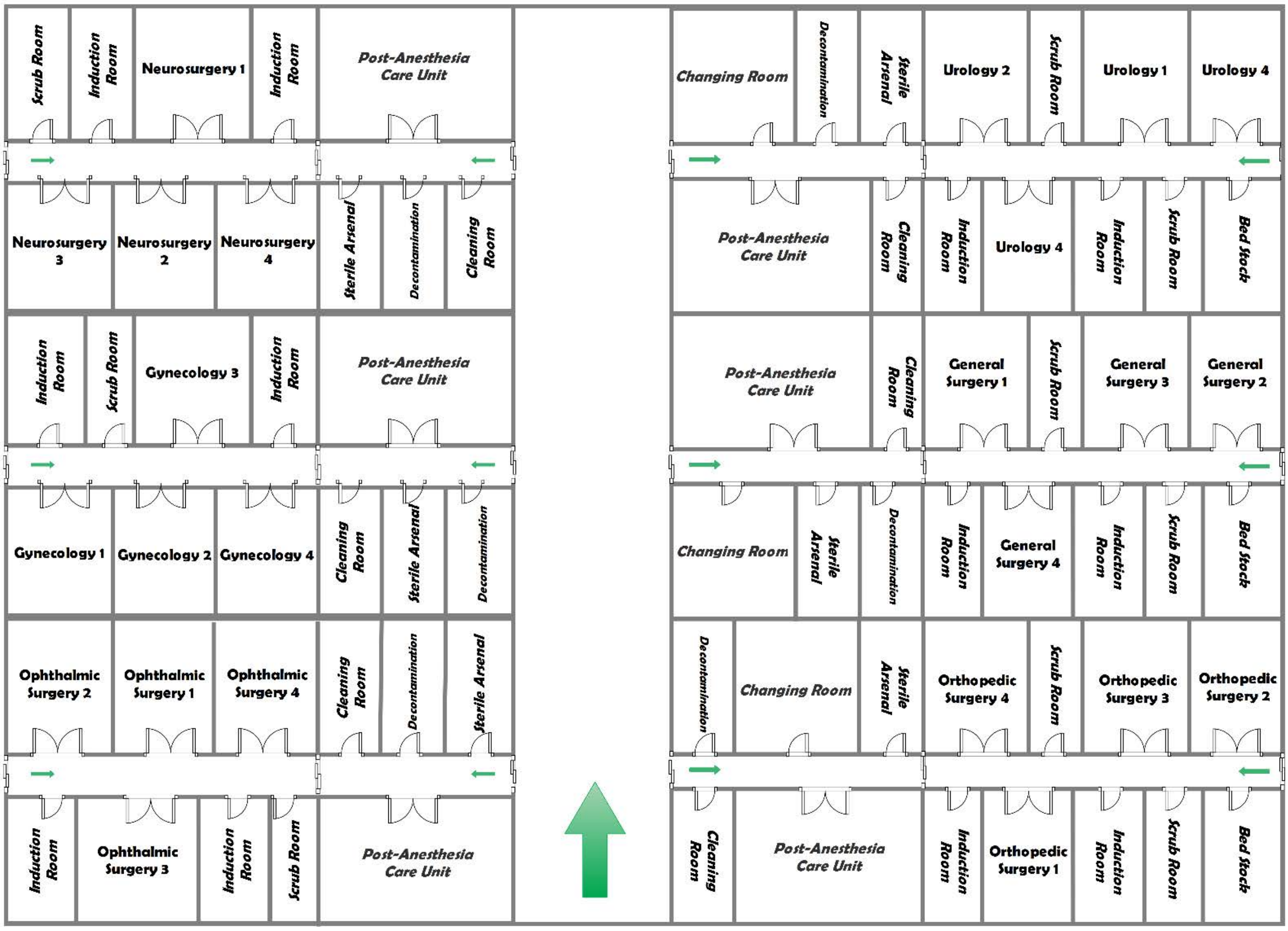


Figure 5.10. Final OT Layout obtained in step 3

Figure 5.9 shows the result of the second step of our proposed algorithm (i.e. Solve

Departments Layout) for an OT problem with 87 facilities while figure 5.10 presents the third step's result for the same problem (i.e. Solve each Department Layout).

5.3.3. Summary

This section presents a novel approach to solve OTLP based on MAS offering hospitals an interactive decision making system to design their OTs. The method has proved to be efficient and effective in solving larger sized problems, which was limited with simple MILP formulation to seventeen facilities.

The unpredictability of population needs has an impact on the required human and material resources and makes the Operating Theater a dynamic environment. Thus, the use of dynamic models is getting more realistic to solve OT Layout Problem (OTLP). The next section will propose an extension of this formulation by applying MAS to the dynamic OTLP.

5.4. Dynamic Operating Theater Layout Problem

Optimal solution with exact method was provided for problems with less than seventeen facilities, which represents the size of one simple department in large hospitals. Based on this evidence, the use of intelligent methods providing solution for larger problem is unavoidable. This section responds to this need by introducing a MAS architecture to solve dynamic OTLP.

5.4.1. Formulation

In this formulation, we introduce a new three levels approach (see Figure 5.11); the first step is to determine the dimensions and orientation of the plant layout or the initial construction site and to divide it into equal size departments corresponding to number of OT specialties. For each department, the second level consists of solving the Dynamic OTLP for different periods of the planning horizon to find the optimal facility locations and orientation using the function (5.4.1). Finally, after choosing the best layout configuration for each department, the third level consists of assigning each one to predetermined location according to its Total Relations Rate (TRR) calculated in equation (5.4.2).

This three-step decision making approach can be best handled through an effective and efficient cooperation of multiple agents in the second level to find the best layout and a negotiation protocol in the third one to allow each agent to occupy the best location in the initial plan. In the next section, we provide a detailed framework using MAS to solve the Dynamic OTLP using this three-level approach.

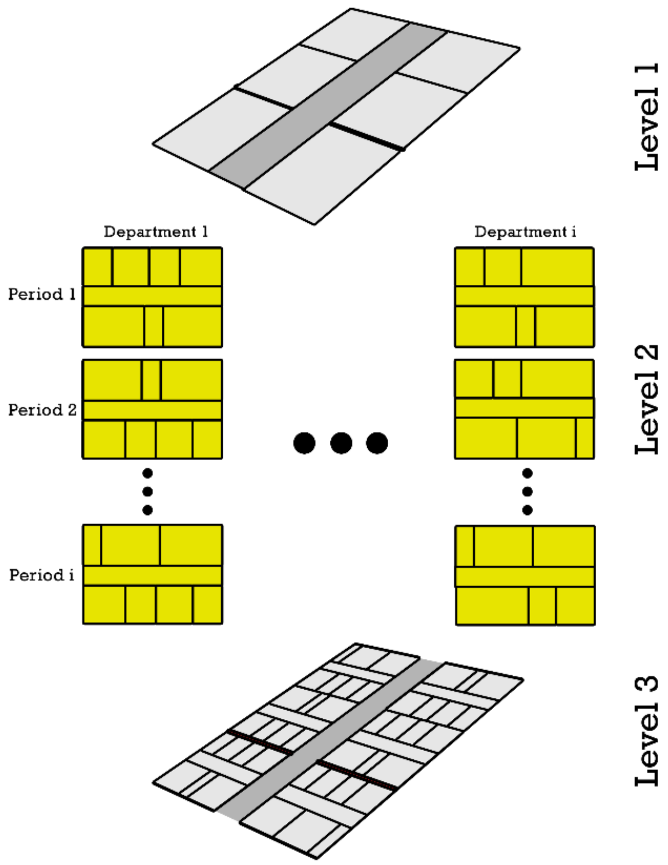


Figure 5.11. The three-level Dynamic FLP approach

The model used in this approach is the same introduced in §3.3 to solve Static OTLP just by considering the period τ_t to all constraints. To solve department layout, we used this MIP formulation while respecting the same assumption:

$$\boldsymbol{min\ F_1} = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{4} \boldsymbol{F_{tijk}(Xp_{tij} + Yp_{tij})(\varphi_{ijk} * \sigma_k)} - \sum_{t=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{R_{ij}\mu_{ijt}} \tag{5.4.1}$$

It is anticipated that the department with higher TRR with others calculated with function (5.4.2) is more interested to occupy a location in the middle of the plan in order to decrease its average distance from other departments. To calculate the TRR we use this formulation:

$$TRR_{hk} = \sum_{i=1}^{N}\sum_{j=1}^{N}(r_{ijhk} + r_{jihk}) \tag{5.4.2}$$

$$h = 1, \dots, D; k = 1, \dots, D$$

While:

- $\boldsymbol{r_{ijhk}}$ is the relationship value which expresses the need for proximity between the facility 'a_i' in the department 'h' and the facility 'a_j' in the department 'k' i.e., if two facilities have a strong positive relationship, they are considered adjacent with a rank of A. The AEIOUX rates are equal to 16, 8, 4, 2, 0 and -2, respectively.

Finally, we use the QAP formulation to calculate the placement of each department while minimizing the traveling costs and maximizing the TRR:

$$\min \sum_{h=1}^{D}\sum_{k=1}^{D}\sum_{m=1}^{D}\sum_{l=1}^{D} C_{hk}D_{hk}X_{hl}X_{km} - \sum_{h=1}^{D}\sum_{k=1}^{D} TRR_{hk}\mu_{hk} \quad (5.4.3)$$

Subject to:

$$\sum_{h=1}^{D} X_{hl} = 1, l = 1, \dots, D \quad (5.4.4)$$

$$\sum_{k=1}^{D} X_{km} = 1, m = 1, \dots, D \quad (5.4.5)$$

- $\boldsymbol{X_{hl}}$: 1 if department h is assigned to location l, 0 otherwise.
- $\boldsymbol{\mu_{hk}}$: is the adjacency coefficient expresses the desirability of locating adjacent departments next to each other. It will be 1 if two departments are fully adjacent, 0.5 if they belong to the same section or 0 if they are located in different sections.
- $\boldsymbol{C_{hk}}$: is the traveling cost between department 'h' and 'k'.
- $\boldsymbol{D_{hk}}$: is the distance between department 'h' and 'k'.

Constraints (5.4.4) and (5.4.5) are to insure that each department is assigned to only one location, and each location contains only one department.

To deal with our problem, we adopt a Master/Slave/Sub-Slave architecture where the Master Processor (MP) is the coordinator of the optimization process, while the Slave Processor (SP) is a worker executing tasks coming from MP and the Sub-Slave Processor (Sub_SP) is a worker executing tasks coming from SP. The main task of a MP is to dispatch parameters and orders for generated SPs, which divide the problem according to the number of periods in the planning horizon and generate Sub_SPs. Once solved, the sub-problem send the answer with objective value to SPs, which negotiate the departments' location. Finally, the MP recuperates the department's layout and location; and gives the final OTLP solution (see Figure 5.12).

The developed MAS tries to solve a multi-department OTLP, where each department is represented as a multi-section layout problem. The developed interaction between agents to solve Dynamic OTLP is described in the following algorithm and can be illustrated in Figure 5.13, Figure 5.14 and Figure 5.15:

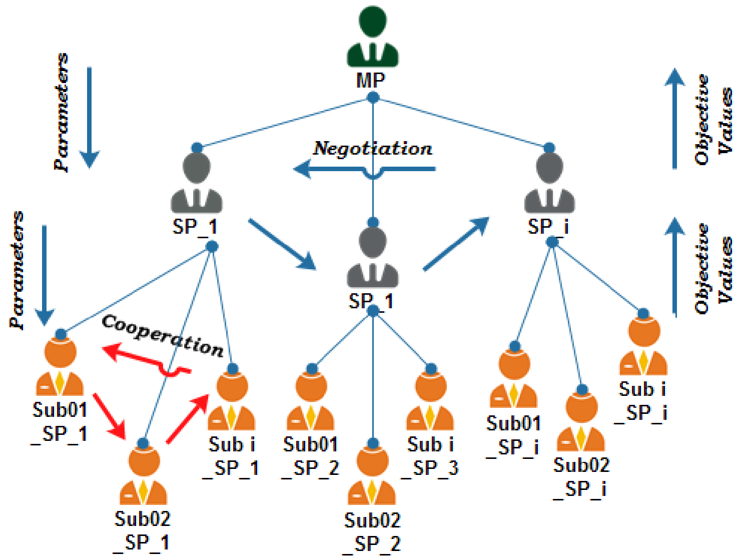


Figure 5.12. The Master/Slave/Sub-Slave architecture

Step 1. Initialization

The initiator launches the mainContainer (JADE platform); setup the graphic Interface (GUI); initialize program parameters and register the MP.

Step 2. Divide the initial OT plan

Once the MP registered, it divides the OT into equal sized sections according to their specialty and as imposed by hospitals managers. The MP sends each department size and orientation to a SP with the required parameters to solve the department layout problem. Once this, the MP is on standby mode waiting for results coming from SPs.

Step 3. Solve each Department Layout

[1] The MP creates and registers a SP for each department;

[2] For each period in the planning horizon, each SP generates a Sub_SP to solve for this period.

[3] Each Sub_SP first receives the instance of the sub-problem from the SP and starts solving MILP formulation using the equation (5.4.1) for the facilities and corridors in the department;

[4] Once all Sub_SPs finish solving the sub-problems in each period for a department ‘d’, agents start cooperation phase by comparing their results in order to choose the best over all periods. The retained solution is then sends with the optimal layout with facilities’ location and orientation to the SP.

Step 4. Solve the whole OT layout

Once all SPs receive answer from their Sub_SPs, they start negotiation phase to determine the best assignment for each department using the QAP formulation (5.4.3) according to their

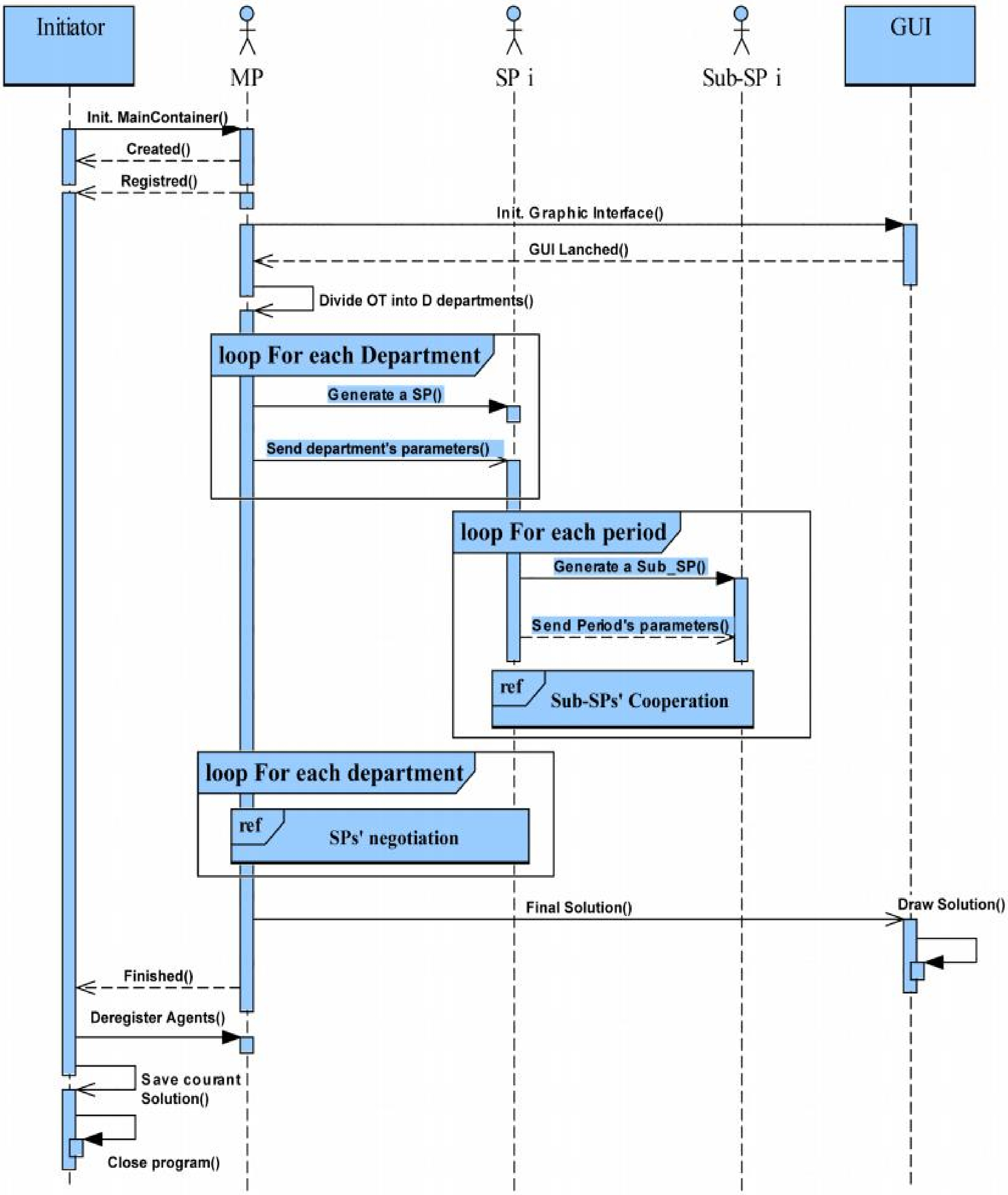


Figure 5.13. The Sequence diagram for the developed algorithm

calculated TRR and send the final solution to the MP. The step stops when the MP collects all solutions.

Step 5. Present the final solution

The result of Sub_SPs' cooperation and SPs' negotiation is a final optimal OT layout. The MP then sends this solution to the GUI that draws it and shows it to the decision makers to validate it.

Step 6. Agents' destruction

Finally, the MP informs the MainContainer about the end of process and destructs all working agents in the JADE platform.

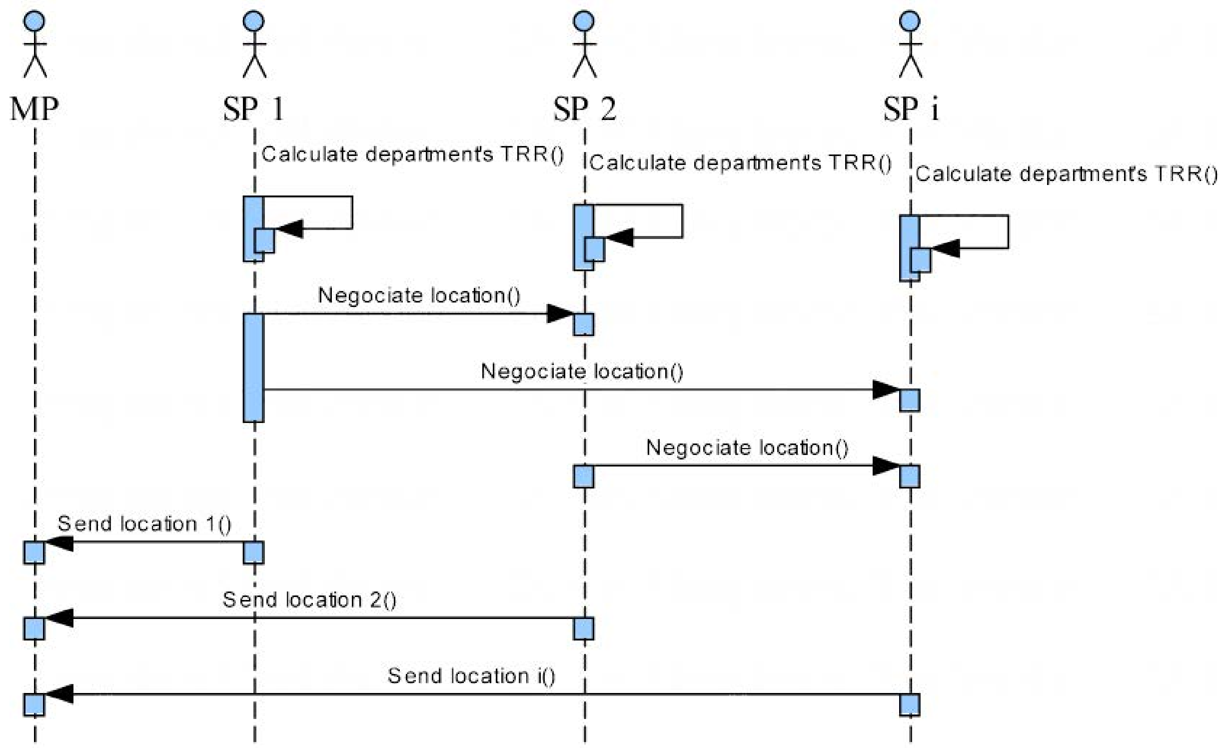


Figure 5.14. Sequence diagram of the Sub-SPs' Cooperation

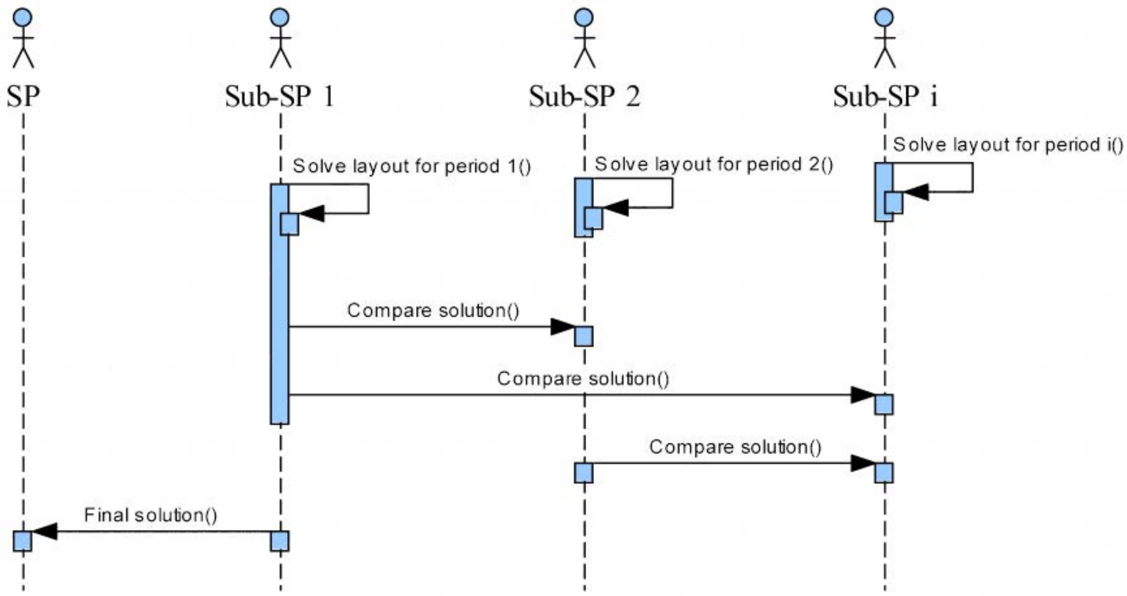


Figure 5.15. Sequence diagram of the SPs' negotiation

In this formulation, we use two agents' communication protocols. The first is cooperation, where agents tried to find a compromising solution for a determined department in such a way that it satisfies the all periods' surgery demand. The second protocol is negotiation, in which agents compete to find a location that satisfies its own benefit. This allow us to exploit agents' properties to find optimal solutions in reasonable time.

5.4.2. Experiments and results

In this section, we validated the proposed MA algorithm based on MILP formulation using a set of previous generated data. We used ILOG CPLEX 12.5 software to solve the model using Windows 7 platform, Intel5® Core ™ i5-3380M CPU@ 2.90GHz and 8Go of RAM.

The algorithm was implemented using JAVA language. We divide the program into three packages: the first contains all the required class to model the JADE agents, the second class models the different OT components (OT, departments, facilities, layout) while the third one is to design the GUI.

Compared to previous obtained results in §4.2.2, this program gives an efficient solution for larger-sized problems in a shorter time. Table 5.2 gives an example of results given by four, six and eight agents in a planning horizon of three, six, nine and twelve periods. As seen, in less than one minute OTLP with 88 facilities can be solved. This offers to architects an interactive and intelligent tool able to design OT while respecting international standards.

The number of facilities we test here is not a limitation, we can make the size larger than 88, while it can increase by increasing the number of agents. Finally, here is an example of obtained solutions: Figure 5.16 shows the final OT layout after the fifth step (i.e. Present the final solution) with 48 facilities.

Number of agents	Number of facilities	Number of periods	Time (sec)		
			Iteration 1	*Iteration 2*	*Iteration 3*
4	48	3	5,14s	4,45s	4,10s
		6	8,50s	8,37s	8,20s
		9	12,73s	12,14s	11,80s
		12	20,21s	17,79s	15,86
6	63	3	7,48s	7,31s	6,84s
		6	14,16s	13,48s	13,33s
		9	23,56s	20,74s	19,76s
		12	28,27s	27,96s	27,24s
8	88	3	10,24s	9,94s	9,51s
		6	21,43s	20,38s	18,71s
		9	30,16s	29,16s	28,01s
		12	44,39s	42,02s	35,48s

Table 5.2. Processing time for different instances sizes of the dynamic formulation

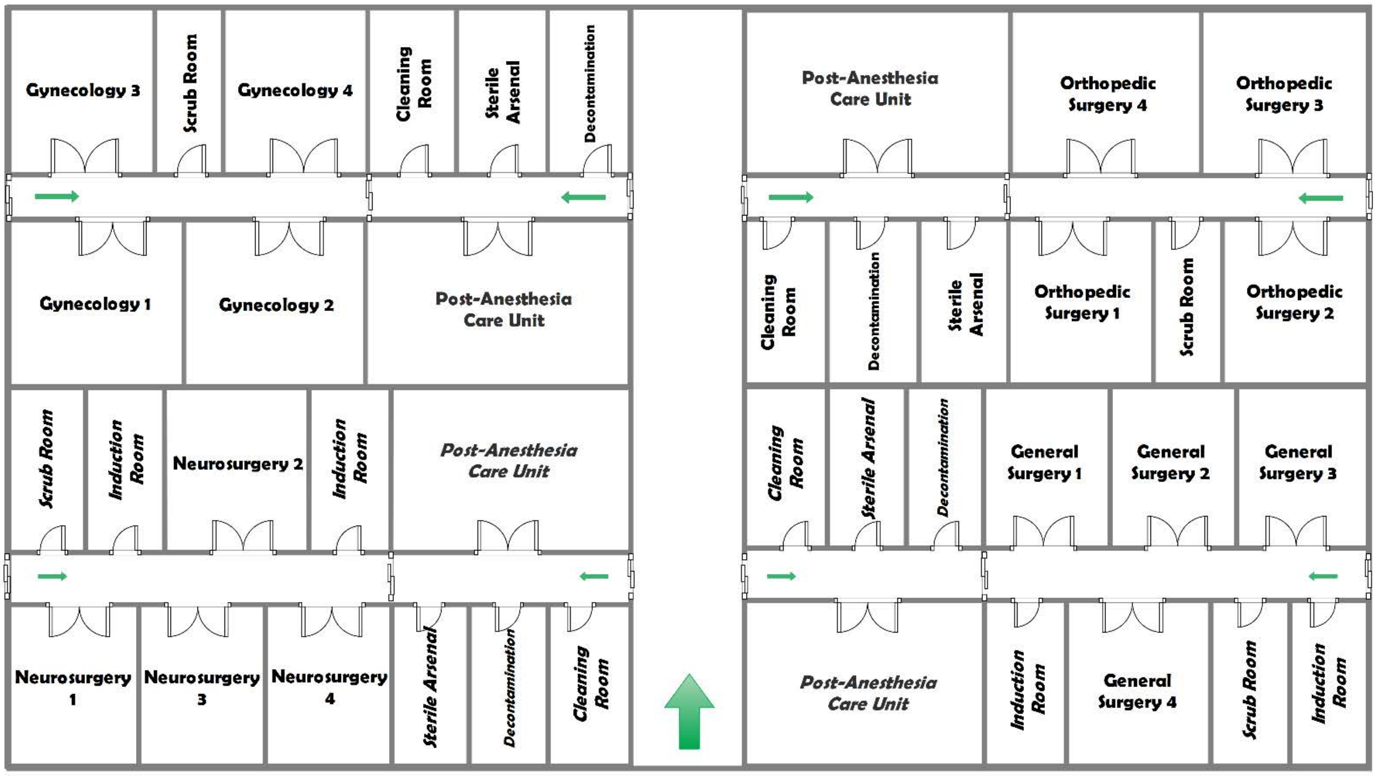


Figure 5.16. Final OT layout with 48 facilities

5.4.3. Summary

This section presents the MAS applied to Dynamic OTLP with the objective to find an optimal solution to the multi-department formulation. The used approach here is FFLP, which consists of finding, in only one single decision, a robust OT layout for the complete planning horizon.

Computational results show the efficiency of MAS to solve complex problems with large-sized instances in different planning horizon sizes. This formulation was able to solve in reasonable time real life OT sizes while considering recommendations of international organizations for designing health care facilities.

5.5. Conclusion

This chapter presents a MAS architecture to solve multi-departments OTLP with static and dynamic environments based on multi-sections formulation in §3.3 and §4.2. The proposed method was tested on large-sized OTLP for which exact methods were unable to give solutions; and which represent a real case of OT in big hospitals.

MAS provides very performant capacity for solving large-sized OTLP using different communication protocols (i.e. cooperation, collaboration and negotiation). In terms of instances' size, we solved problems with 162 for static OTLP and 88 for dynamic OTLP facilities but not necessarily limited to these number. In terms of processing time, final solutions was obtained in reduced time compared to those obtained with exact methods.

In further works, MAS will be applied to multi-floors and multi-rows formulation with static and dynamic environment. Furthermore, we did not investigate all MAS features, which could perform our approach and be an additional contribution to OTLP.

5.6. Résumé

Ce chapitre présente une architecture de systèmes multi-agent pour résoudre le problème de disposition du bloc opératoire pour une structure à plusieurs départements sous un environnement statique et dynamique basé sur la formulation de structures à plusieurs sections vue dans §3.3 et §4.2. La méthode proposée a été testée sur des OTLP de grandes tailles pour lesquels les méthodes exactes étaient incapables de donner des solutions ; et qui représentent un cas réel de bloc opératoire dans les grands hôpitaux.

Le système multi-agent offre une capacité très performante pour résoudre des OTLP de grandes tailles en utilisant des protocoles de communication différents (à savoir la coopération, de collaboration et de négociation). En termes de tailles d'instances, nous avons résolu des problèmes avec 162 installations pour l'OTLP statique et 88 installations pour l'OTLP dynamique mais nous ne sommes pas nécessairement limités à ces nombres. En termes de temps de traitement, des solutions ont été obtenues dans un temps réduit par rapport à celles obtenues avec des méthodes exactes.

Dans les travaux à venir, les systèmes multi-agent seront appliqués à des structures à plusieurs étages et à plusieurs rangées sous environnement statique et dynamique. En outre, nous n'avons pas encore exploré tous les atouts et les caractéristiques des systèmes multi-agent qui pourraient améliorer notre approche et ajouter d'autres contributions supplémentaires pour la résolution de l'OTLP.

Chapter VI.
Particle Swarm Optimization for Operating Theater Layout Problem

6.1. Introduction

Resolution of the facility layout problem (FLP) is based either on exact methods to reach optimal solutions, or heuristics and meta-heuristics to get near-optimal solutions. Exact methods cannot give optimal solutions for larger-size instances due to the NP-completeness of such FLPs. For this reason, heuristics or meta-heuristics are the most appropriate methods for solving the large size of this problem in a reasonable computational time to offer near optimal solutions. These meta-heuristics include Genetic Algorithm (GA), Tabu Search (TS), Simulated Annealing (SA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO).

Before going more into technical details of the PSO, we provide in the following a description of some of these metaheuristics and a comparison between them to argue our choice.

6.1.1. Genetic Algorithm

As all intelligent methods have been inspired from natural phenomena or from living beings' behavior, GA has been developed by Holland (1975) based on the natural process of evolution in human. GA starts with a set of initial solutions named chromosomes generated randomly, heuristically or by a combination of both. This set of solutions is known as the population in which each chromosome consists of genes.

In GA, four genetic operations are used to generate new chromosomes (called children). The ***selection*** is a process in which two individuals called parents are selected from the current population for the next genetic operations. The ***crossover,*** are operations executed in order to generate new children from the two parents selected in the selection phase, while the ***mutation*** and the ***inversion*** are used to modify an existing chromosome.

The selection of chromosomes to crossover and mutate is based on their fitness function using an iterative search process. The chromosomes evolve through successive iterations called ‘generations’ to create a new population by applying the crossover and mutation operators. The process is repeated until a specific stopping criterion is met.

Here we present the algorithm of the Genetic Algorithm (Sean [2014]):

```
popsize ← desired population size
P ← {}
for popsize times do
      P ← P ∪ {new random individual}
Best ← □ (null)
repeat
      for each individual Pi ∈ P do
            AssessFitness(Pi)
            if Best = □ or Fitness(Pi) > Fitness(Best) then
                  Best ← Pi
      Q ← {}
      for popsize /2 times do
            Parent Pa ← SelectWithReplacement(P)
            Parent Pb ← SelectWithReplacement(P)
            Children Ca, Cb ← Crossover(Copy(Pa), Copy(Pb))
            Q ← Q ∪ {Mutate(Ca), Mutate(Cb)}
      P ← Q
until Best is the ideal solution or we have run out of time
return Best
```

6.1.2. Tabu Search

TS was introduced by Glover (1989) as a deterministic local search strategy in order to solve combinatorial optimization problems. TS uses a short-term memory called the ‘tabu list’ to store recently visited solutions to avoid short-term cycling. At each iteration, a local search process is started to produce a new solution S’ in the neighborhood of current one S where the just found solutions are called ‘taboo’ (i.e. forbidden to be visited), are stored in the tabu list.

Thus if S′ is better than S, S’ it is considered as the best current solution in order to find the optimal one.

Typically, the search stops after a determined number of iterations or a maximum number of consecutive iterations without any improvement to the best-known solution. Here we present the Tabu Search algorithm from Sean (2014):

```
l ← Desired maximum tabu list length
n ← number of tweaks desired to sample the gradient
S ← some initial candidate solution
Best ← S
L ← {} a tabu list of maximum length l (Implemented as first in,
first-out queue)
Enqueue S into L
repeat
      if Length(L) > l then
            Remove oldest element from L
      R ← Tweak(Copy(S))
      for n - 1 times do
            W ← Tweak(Copy(S))
            if W ∉ L and (Quality(W) > Quality(R) or R ∈ L) then
                  R ← W
      if R ∉ L and Quality(R) > Quality(S) then
            S ← R
            Enqueue R into L
      if Quality(S) > Quality(Best) then
            Best ← S
until Best is the ideal solution or we have run out of time
return Best
```

6.1.3. Simulated Annealing

SA is a randomized local search procedure, introduced by Kirkpatrick et al. (1983) where the idea was inspired from the annealing technique used by the metallurgists to find low-energy states of solids. SA consists of carrying a material at high temperature, then lowering this temperature slowly, while avoiding the metastable structures, characterizing the local minima of energy. Thus, if the initial temperature is not high enough or cooling process is very quick, the solid will have many defects or imperfections.

SA makes a mix of random selection and of a hill-climbing search. In fact, exploring research space, choosing neighbors and deciding whether to visit that neighbor are done randomly. If a visited neighbor provides an improvement to the actual solution, the procedure continues from this neighbor. Otherwise, it still could be chosen according to the Metropolis criterion with a probability depending on a parameter called the “temperature". By accepting these worse solutions, SA has the capability of jumping out of local optima.

Here we introduce the Simulated Annealing algorithm presented in Sean (2014):

```
t ← temperature, initially a high number
S ← some initial candidate solution
Best ← S
repeat
	R ← Tweak(Copy(S))
	if Quality(R) > Quality(S) or if a random number chosen from 0
	to 1 < e^((Quality(R)−Quality(S))/t) then
		S ← R
		Decrease t
	if Quality(S) > Quality(Best) then
		Best ← S
until Best is the ideal solution, we have run out of time, or t ≤ 0
return Best
```

6.1.4. Ant Colony Optimization

ACO is population-oriented metaheuristic introduced by Dorigo (1991) for hard combinatorial optimization problems, being inspired by the social behavior of real ants to find the shortest path from the nest to the food source while using pheromone trails as a way to communicate. In ACO, ants use the pheromone trails during the algorithm’s execution as a distributed numerical information to reflect their search experience.

Ants construct solutions by searching for food in a randomized and greedy way. This stochastic element in the ACO allows ants to build a diversity of solutions. When an ant finds a food source, it evaluates it quality and it quantity and deposits a chemical pheromone trail on the ground when coming back to the nest to mark some favorable path. Thus, the shortest path concentrates the higher quantity of pheromone and attracts more ants and should guide them to the food source.

Artificial ant colonies exploited this characteristic of real ant colonies to build solutions to an optimization problem and exchange information on their quality through a communication scheme that is reminiscent of the one adopted by real ants (Dorigo [2006]). Different types of the ACO algorithms have been proposed such as ant system (AS), elitist AS, ant-q, ant colony system, max–min ant system (MMAS), rank-based AS, ants, best–worst ant system, and hyper-cube AS.

Here we present an abstract of the Ant Colony Optimization algorithm (ACO):

```
C ← {C1, ..., Cn} components
popsize ← number of trails to build at once
p⃗ ← ⟨p1,...,pn⟩ pheromones of the components, initially zero
Best ← □
repeat
    P ← popsize trails built by iteratively selecting components
    based on pheromones and costs or values
    for Pi ∈ P do
        Pi ← Optionally Hill-Climb Pi
        if Best = □ or Fitness(Pi) > Fitness(Best) then
            Best ← Pi
    Update p⃗  for components based on the fitness results for each
    Pi ∈ P in which they participated
until Best is the ideal solution or we have run out of time
return Best
```

6.1.5. Particle Swarm Optimization

The PSO method belongs to the field of swarm intelligence, which has involved important attention over the past decade. Originally designed and developed by Kennedy and Eberhart (1995) as a global optimization technique. They described a new field of optimization methods inspired by the social behaviors of animals without leader such as birds flocking or fish schooling. These natural swarms explore the searching area in a random way to find food and follow one of the group members with the nearest position to a food source. This latter will inform the other members about the potential solution in such a way to push them to change their trajectory simultaneously to that place. This process will be repeated until the discovery of the food source.

To find optimal value for an optimization problem, the PSO algorithm follows the behavior of this animal society. PSO consists of a swarm of particles, where a particle represents a candidate solution to the problem and has a velocity (i.e. the traveling distance and direction of each particle in each iteration), a location in the search space and a memory to remember its previous best position.

More details of the PSO will be giving in the next section, while the algorithm of PSO described in Sean (2015) is presented here:

```
swarmsize ← desired swarm size
α ← proportion of velocity to be retained
β ← proportion of personal best to be retained
γ ← proportion of the informants' best to be retained
δ ← proportion of global best to be retained
ε ← jump size of a particle
P ← {}
for swarmsize times do
      P ← P ∪ {new random particle x⃗ with a random initial velocity
      v⃗}
Best⃗ ← □
repeat
      for each particle x⃗ ∈ P with velocity p⃗ do
            AssessFitness(x⃗)
            if Best⃗ = □ or Fitness(x⃗) > Fitness(Best⃗) then
                  Best⃗ x⃗
      for each particle x⃗ ∈ P with velocity v⃗ do
            x⃗* ← previous fittest location of x⃗
            x⃗+ ← previous fittest location of informants of x⃗
            x⃗! ← previous fittest location any particle
            for each dimension i do
                  b ← random number from 0.0 to β inclusive
                  c ← random number from 0.0 to γ inclusive
                  d ← random number from 0.0 to δ inclusive
                     v_i ← αv_i + b(x⃗*_i − x_i) + c(x⃗+_i − x_i) + d(x⃗!_i − x_i)
      for each particle x⃗ ∈ P with velocity v⃗ do
```

$$\vec{x} \leftarrow \vec{x} + \varepsilon\vec{v}$$

```
until Best is the ideal solution or we have run out of time
return Best
```

6.1.6. Summary

Metaheuristics are numerous and the development difficulty differs from one to another. This section presents a non-exhaustive list of existing metaheuristics and gives the algorithm to develop each one. The choice of the adequate metaheuristic to use depends of optimization problem, it complexity, the constraints type, etc.

Moslemipour el al. (2012) presented a comparison between well-known metaheuristics and exposed the advantages and disadvantages of each one. In the following, an abstract of this comparison is presented (Table 6.1):

Approach	Advantages	Disadvantages
GA	The ability of: 1. Solving different kinds of COPs 2. Finding a global best solution 3. Combining with other algorithms	1. Very slow 2. Finding sub-optimal solution 3. Converging even to local optima is not guaranteed 4. The crossover and mutation rates affects the stability and convergence 5. Dependency of the evaluation performance on the gene coding method
TS	Using the flexible memory to retain the history of the search process	The obtained solution is not necessarily an optimal solution
SA	1. Low computational time 2. Free of local optima 3. Easy for implementation 4. Convergent property	Dependency of the solution quality on the maximum iteration number of the inner loop (cooling schedule) and the initial temperature
ACO	Scalability, robustness and flexibility in dynamic environments	1. It is not easy for coding 2. Parameters initializations by trial and errors or at random
PSO	1. Easy to implement, fast and cheap 2. Having few parameters to adjust 3. Efficient global search approach 4. Having simple structure 5. Less dependent of a set of initial points	Having weak local search ability

Table 6.1. Advantages and disadvantages of intelligent approaches (Moslemipour el al. [2012])

6.2. Particle Swarm Optimization

The above advantages provided by PSO algorithm invite us to use it at first for solving OTLP with a perspective to explore other algorithms in order to make a comparison. This section is provided to present the PSO in details, the basic concepts and different structures and variants.

6.2.1. The basic concepts of PSO algorithm

As mentioned above, PSO was initially introduced in 1995 as a global optimization technique by Kennedy and Eberhart. The PSO algorithm is based on a set of autonomous entities called particles that are originally arranged randomly and homogeneously in the search space. Each particle is a potential solution to the problem, has a memory to remember its previous best position and has the ability to communicate with other particles in its entourage. This entourage is historically called the neighborhood and may be the entire population or some subset of it.

A swarm consists of N particles flying around in a multi-dimensional search space looking for the global optimum. Furthermore, the movement of a particle is influenced by the following three components:

- The inertia component: It represents the tendency of the particle to follow its current direction of travel.
- The cognitive component: It represents the tendency of the particle to move towards the best position in which it has already passed.
- The social component: It represents the tendency of the particle to rely on the experience of other particles and to head for the best position already achieved by its neighbors.

6.2.2. Swarm topologies

When moving, each particle maintains a sort of interconnection with other particles giving it the access to the information of all other members of the population. These interconnections can be described by the following swarm topologies (see Figure 6.1):

- The *star topology*: or the ***gBest*** topology, which is a fully connected graph where each particle is connected to all other particle. The advantage of this topology is its fast convergence compared to other topologies, even if this feature can lead to be trapped in local minima.
- The *ring topology*: or the ***lBest*** (local best) topology, where each particle is interconnected with only its two adjacent neighbors in a way that when a particle finds a better solution, it informs its two adjacent neighbors which also inform their adjacent

neighbors, until the information reaches the last particle. The disadvantage of this topology compared to others, is its slow convergence due to the slow spread of the best-founded solution. However, this topology offers a larger exploration of the search space and typically ends up at a better optimum.

- The *wheel topology*: in which only one particle (called a focal particle) is interconnected to the other particles. This case is similar to the master/slave architecture provided in MAS in chapter §5 where all the information is communicated through this coordinator particle. In this topology, the focal particle compares the best result of all other particles in the swarm, adjusts its position around the best result particle and informs all particles about its new position.

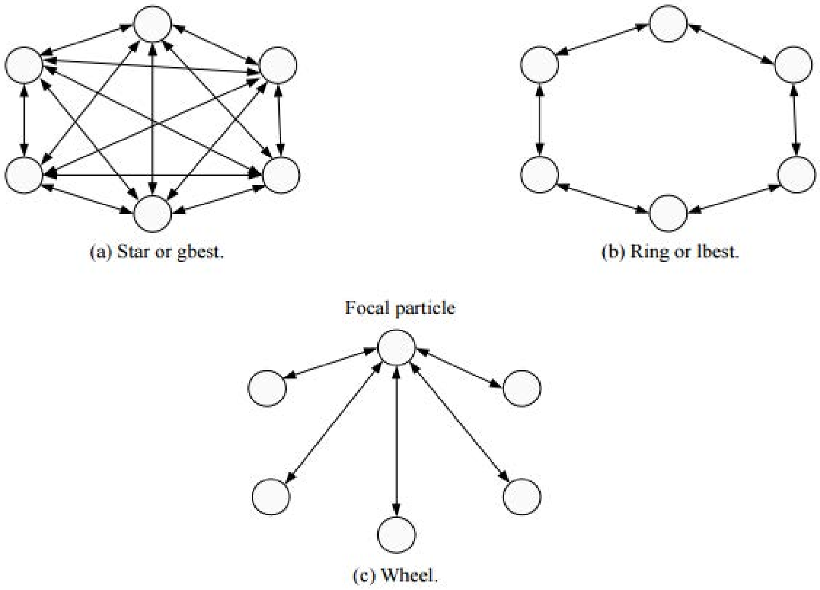


Figure 6.1. Different swarm topologies

6.2.3. Particle Behavior

Each particle is represented by its location in space: $\vec{x} = \langle x_1, x_2, \dots \rangle$ and its velocity: $\vec{v} = \langle v_1, v_2, \dots \rangle$, which define the direction and the distance the particle should travel at each iteration. In the starting phase of PSO, the location and velocity of each particle are randomly initialized. While after each iteration, each particle adjusts its position $\vec{x}_i$ and velocity $\vec{v}_i$ in the search space, based on its personal best *Pbest* result and the global best result found noted *Gbest* by any other particle in its neighborhood. Equations (6.2.1) and (6.2.2) are used to update the particle's velocity in the two topology cases: star and ring (gBest and lBest) respectively, while (6.2.3) is used to update the position of each particle after each iteration.

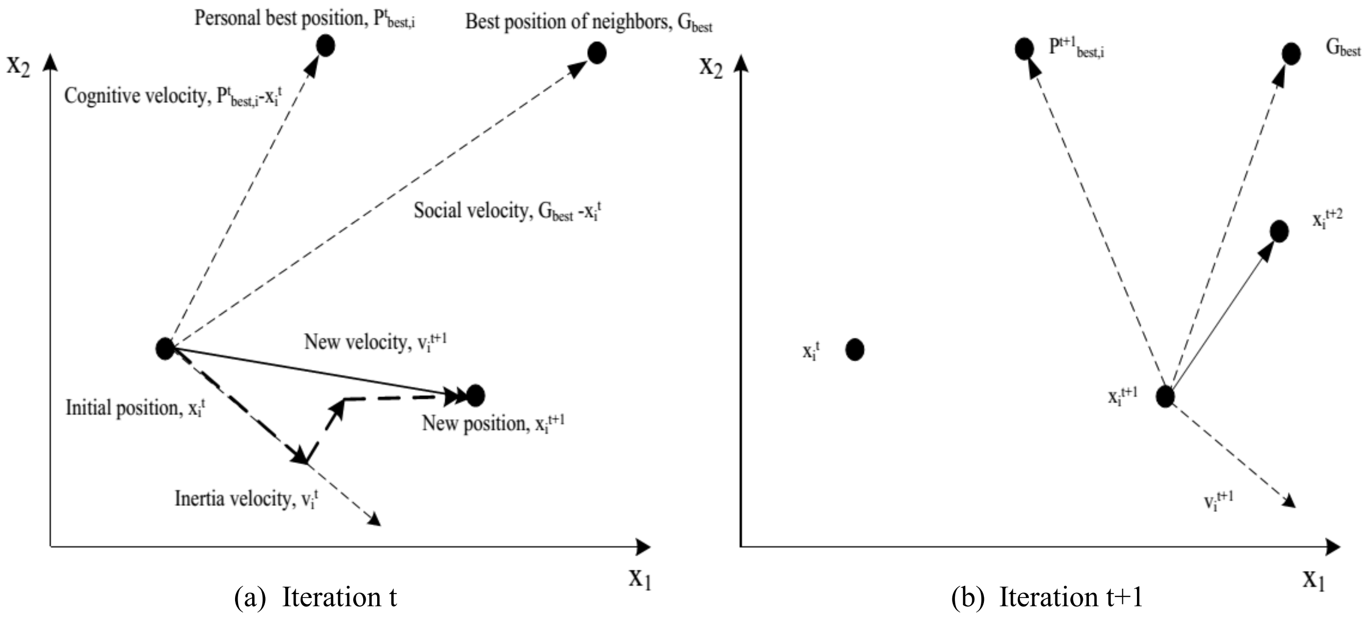
$$V_i^{t+1} = \omega V_i^t + \beta_1 c_1 \left(Pbest_i - x_i^t\right) + \beta_2 c_2 (Gbest_i - x_i^t) \tag{6.2.1}$$

$$V_i^{t+1} = \omega V_i^t + \beta_1 c_1 \left(Pbest_i - x_i^t\right) + \beta_2 c_2 \left(Lbest_i - x_i^t\right) \tag{6.2.2}$$

$$x_i^{t+1} = x_i^t + V_i^{t+1} \tag{6.2.3}$$

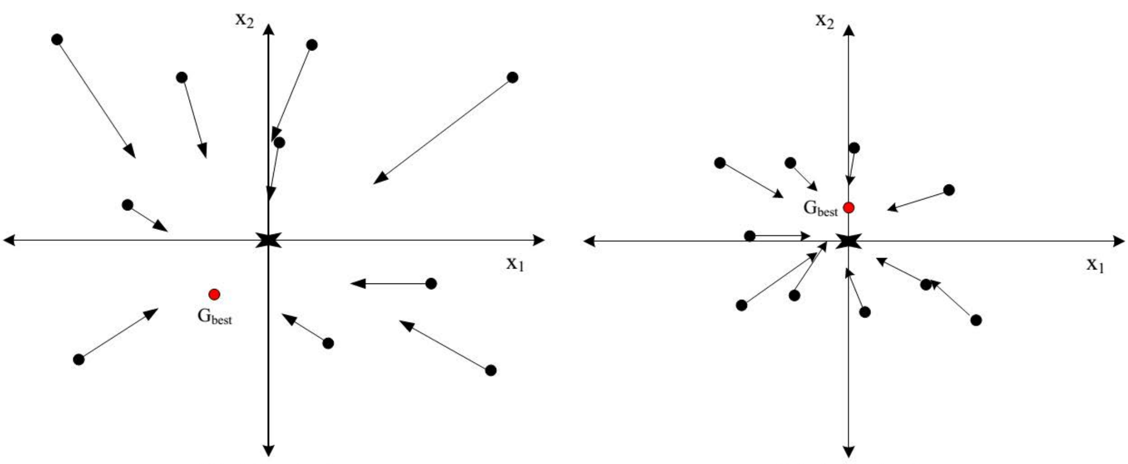
Where ω is the inertia weight, c_1 and c_2 are the acceleration coefficients, while β_1 and β_2 are two vectors of random numbers between 0 and 1.

Figure 6.2 shows the way a particle changes it position from iteration t to t+1 in a two-dimensional space search to reach the global best position. While Figure 6.3 shows the position update for all particles in a two-dimensional search space towards the gBest PSO, where the space center point illustrates the optimum position and the red point represents the gBest solution.



(a) Iteration t (b) Iteration t+1

Figure 6.2. Velocity and position update for a particle in a two-dimensional search space



(a) at time t=0 (b) at time t=1

Figure 6.3. Velocity and Position update for Multi-particle in gbest PSO

6.2.4. The gBest and lBest

As mentioned above, the gBest is presented as a method using the star topology where each particle has the entire swarm as a neighborhood and where the position of each particle is influenced by the global best position in the entire population. On the other hand, lBest method uses the ring topology where each particle is connected to its two immediate neighborhoods.

Engelbrecht (2013) presented a detailed comparison between these two concepts along with the advantages and disadvantages of each one:

- gBest converges faster than lBest
- gBest has more chance to be trapped in a local minima than lBest
- gBest PSO is more suitable to solve unimodal problems than multimodal ones
- gBest PSO does not perform well in case of non-separable problems
- Increasing the size of neighborhoods deteriorates performance. Thus, lBest finds accurate solutions than gBest for the majority of the problems.

Figure 6.4 and Figure 6.5 are reported from Talukder (2011) to show the difference between the two methods' algorithms.

6.2.5. PSO Parameters

The main parameters of the PSO algorithm influencing its performance are the inertia weight ω, the acceleration coefficients c_1 and c_2, the maximum velocity V_{max}, the swarm size S and the stopping criteria. The setting of these parameters has an important impact on the efficiency of the PSO and determines how it optimizes the search-space.

- The *inertia weight* is used as a coefficient to weight the contribution of the velocity at iteration t when calculating the new velocity at iteration t+1. The inertia weight controls both the momentum of the particle and the exploration of the search space. The setting of this coefficient can be either a fixed or a dynamically changing value. Furthermore, it is recommended to set its value in the interval $]0,1[$ to avoid losing knowledge of the past velocity when exploring the search space if ω=0 and to avoid having a quick increase of the velocity over time if ω=1, which involves the swarm divergences because particles will hardly change their direction to move back towards optimum.
- The *swarm size* called also the population size is the number of particles in the swarm. It essentially depends on two parameters: the search space size and the ratio between the executing machine performance and the maximum searching time. When setting a big number of particles, larger parts of the search space can be covered per iteration and thus the number of iterations needed to obtain a good solution is reduced. However, this increases the computational complexity per iteration, and consumes more time.

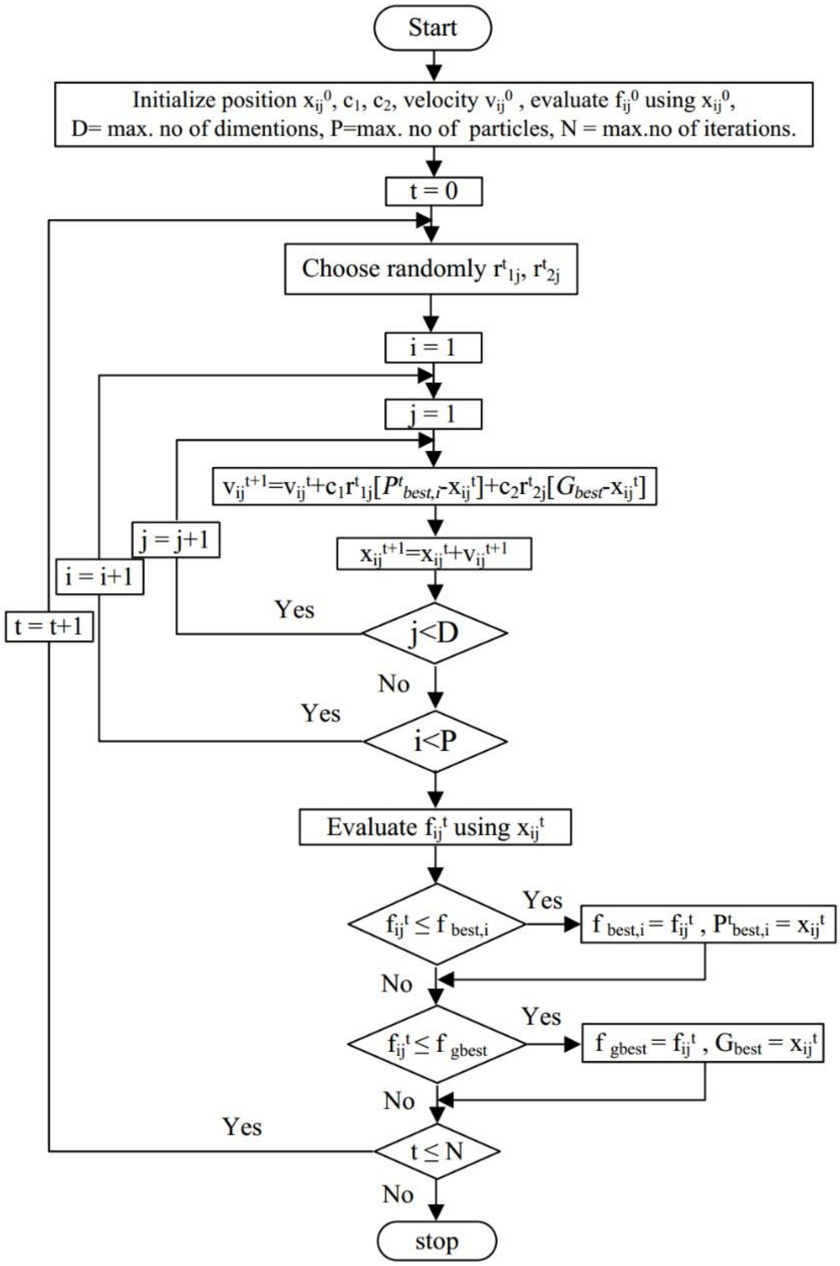


Figure 6.4. The gBest algorithm

- The *stopping criteria* are used as an exit gate to ensure stopping the execution of the algorithm at some point because the convergence to the global optimal solution cannot be guaranteed in all cases even if the experiments indicate the high performance of the method. Thus the algorithm must run until one of the following convergence criteria have not been met:
    - the maximum number of iteration is reached;
    - no improvement is observed over a number of iteration;
    - the velocity is close to zero;
    - the normalized swarm radius is close to zero;
    - the objective function slope is approximately zero;

    - the actual fitness is acceptable.
- The *maximum velocity* determines the maximum change one particle can have in each iteration and restricts its step size within the defined boundary in order to prevent the particles from moving quickly in the search space and eventually miss the optimum and to create a better balance between global exploration and local exploitation. Thus, to define the V_{max}, it is important to avoid setting a very high value that create irregular particle movement, missing a good solution; or to define a very small value that limits the particle’s movement and may run the risk of not reaching the optimal solution.

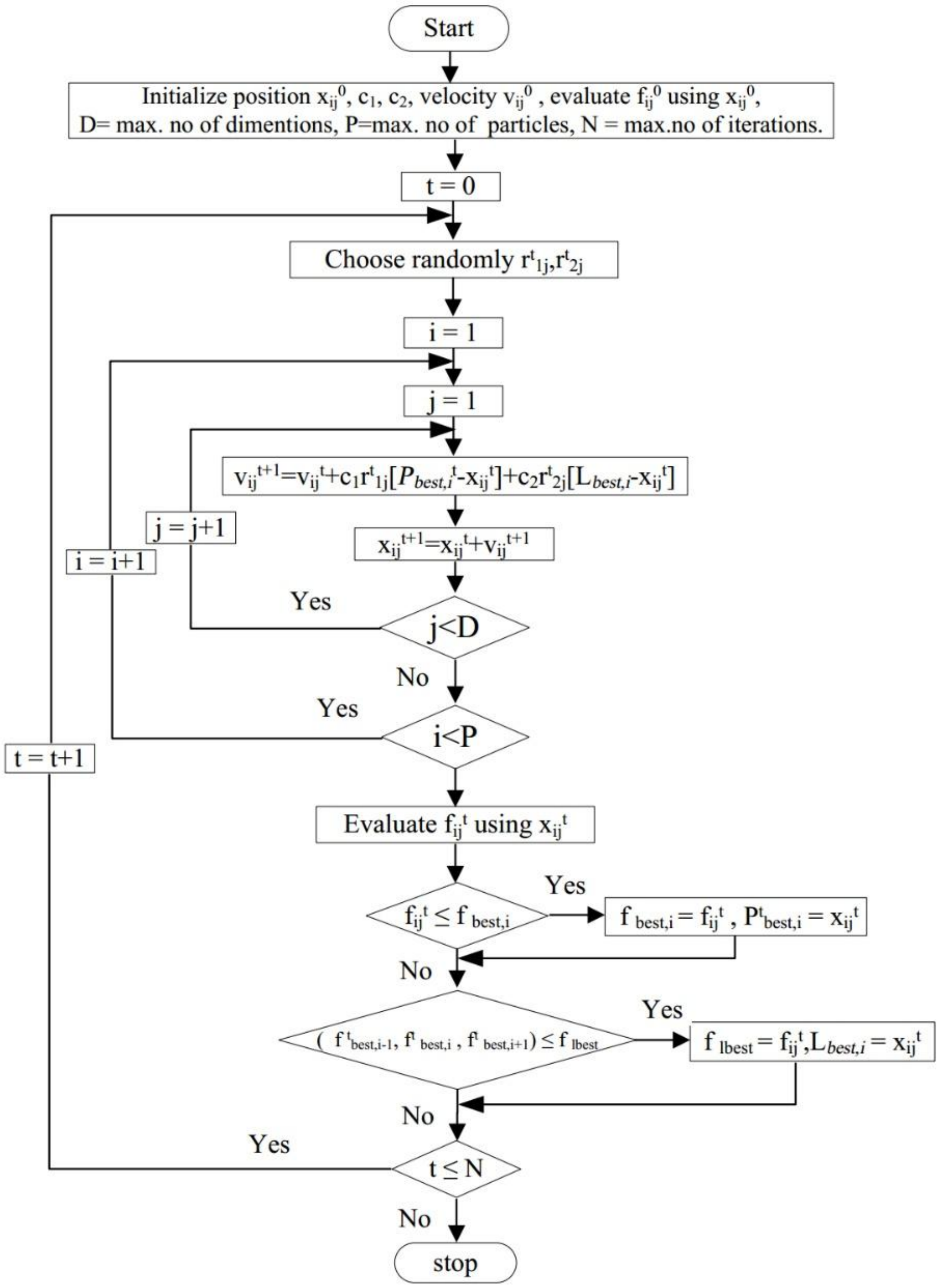


Figure 6.5. The lBest algorithm

- The *acceleration coefficients* c_1 and c_2 are used to maintain the stochastic influence of the cognitive and social components of the particle's velocity, respectively. The constant c_1 is called "self-confidence" since it expresses the experience of a particle and how much confidence a particle has in itself, while c_2 is also called "swarm confidence" and expresses how much confidence a particle has in its neighbors taking into account the motion of all the particles in the preceding iterations. When setting these parameters, there are some properties to consider:
  - When $c_1 = c_2 = 0, V_i^{t+1} = V_i^t$ i.e. particles continue moving with the same speed until they hit the defined search space boundary;
  - When $c_1 > 0$ and $c_2 = 0$, all particles are independent since each one moves according to its self-experience;
  - When $c_1 = 0$ and $c_2 > 0$, all particles will be attracted to the gBest point of the entire swarm.
  - When $c_1 = c_2$ particles are attracted towards the average of pBest and gBest points.
  - From the empirical research in the literature, it has been recommended to set these parameters $c_1 = c_2 = 2$.

6.2.6. Summary

The PSO method belongs to the field of swarm intelligence inspired by the social behaviors of animals without leader such as birds flocking or fish schooling. PSO consists of a swarm of particles, where a particle represents a candidate solution to the problem and has a velocity, a location in the search space and a memory to remember its previous best position.

This section discussed briefly the basic concepts of the PSO algorithm, the different neighborhoods' topologies with the algorithm and formulation related to each one, the principle of particles' moving in the search space and the major parameters having a large influence on the PSO results.

The next section present our developed PSO algorithm to deal with the multi-departments OTLP adopting the continuous representation, the way to encode each particle, the used mathematical model to solve the problem and different experiments and results.

6.3. Static Operating Theater Layout Problem

PSO has been developed to multi-departments OTLP in which departments are formulated as multi-row problems. The application of PSO to FLP in the literature often deals with discrete representations, i.e. the layout is divided into equal sized locations in a way that the problem is reduced to allocate facilities to these locations. In our formulation, we consider continuous representation involving the conception of a novel method to code particles of our PSO.

In this section, we will firstly expose the mathematical model. Then, the developed PSO method will be presented and finally numerical results will be illustrated as well as comparison with exact method's results.

6.3.1. Mathematical formulation

Assumptions

- The facilities have rectangular shapes with non-equal areas of known dimensions;
- The locations of the department and rows are fixed a priori;
- Transition between departments can only occur through corridors;
- Each facility is located in a row parallel to the horizontal axis;
- All facilities located on the same row have the same y-axis value;
- Corridors are located parallel to the horizontal axis and their location is known;
- All rows have the same width and the same length;
- Each department is composed of two rows (side 1 and side 2);
- The number of corridors and rows are known;
- Inter-facility distance is computed using Manhattan norm while travel is ensured by corridors;
- A vertical corridor is considered to allow routing between different rows; we call it 'Central Corridor' and assign it the index c = 1.

Sets

Let N= $\{a_i\,;\, i=1,2,\ldots,n\}$ be the set of n facilities;

Let K= $\{e_k\,;\, k=1,\ldots,4\}$ be the set of k entity types;

Let C= $\{\theta_c;\, c=1, 2,\ldots,r\}$ be the set of corridors;

Let R= $\{r_r;\, e=1, 2,\ldots,v\}$ be the set of rows;

Let D= $\{\delta_d\,;\, s=1, \ldots, R\}$ be the set of departments;

Let U_i be a set of a single element denoting the row to which facility a_i is belonging.

Parameters

l_c, d_c:	Dimensions of corridor θ_c
L_r, D_r:	Dimensions of row r_r
F_{ijk}:	Number of trips between facility a_i to a_j made by an entity type e_k
φ_{ijk}:	Moving difficulty between facility a_i to a_j made by an entity type e_k
σ_k:	Cost of horizontal traveling by entity e_k

$X_{max:}$	Maximum length of OT ;
Y_{max}:	Maximum width of OT ;
x_c ,y_c :	coordinates of the geometrical center of corridor θ_c
x_r ,y_r :	coordinates of the geometrical center of row r_r
R_{ij}:	Desirable relationship value between facility a_i to facility a_j;
ρ_1, ρ_2:	Weights for each sub-objective function.
H_c	Distance between two successive corridors

Decision Variables

μ_{ij} $\begin{cases} 1 \text{ if } a_i \text{ and } a_j \text{ are fully adjacent} \\ 0 \text{ otherwise} \end{cases}$

z_{ij}^{x} $\begin{cases} 1 \text{ if facility } a_i \text{ is strictly to the right of facility } a_j \\ 0 \text{ otherwise} \end{cases}$

v_{ic} $\begin{cases} 1 \text{ if facility } a_i \text{ is adjacent to corridor } \theta_c \\ 0 \text{ otherwise} \end{cases}$

o_{ij} $\begin{cases} 1 \text{ if facilities } a_i \text{ and } aj \text{ are adjacent to the same corridor } \theta_c \\ 0 \text{ otherwise} \end{cases}$

x_i , y_i:	x and y coordinates of the geometric center of gravity facility a_i
l_i:	x-length of facility a_i depending on whether α_i or β_i is parallel on x-axis
d_i:	y-length of facility a_i depending on whether α_i or β_i is parallel on y-axis
Xp_{ij}:	x-distance between facility a_i and a_j in the same floor
Yp_{ij}:	y-distance between facility a_i and a_j in the same floor
Xc_i:	x-distance between facility a_i and central corridor θ_c
Y_{c1c2}:	Vertical distance between corridor θ_{c2} and θ_{c2}

Objective functions

The goal of the OTL problem is to provide the best placement of the facilities within the available space. This objective function is computed as the sum of two weighted sub-functions in (6.3.1). The first sub-function minimizes the material handling costs. It is proportional to the rectilinear distance, travel frequency, trip difficulty rating, and baseline travel cost, while the second sub-function maximizes the desired closeness-rating factor based on international standards. The qualitative factors easily determine the closeness-rating chart. The chart is essentially a grid that qualitatively evaluates the desired closeness between any two facilities.

The rates determine the strength of the closeness as follows: necessary (A), very important (E), important (I), ordinary importance (O), un-important (U) and undesirable (X).

$$\boldsymbol{min\ F} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{4} \boldsymbol{F_{ijk}\ (Xp_{ij} + Yp_{ij} + Y_{c1c2})\varphi_{ijk} * \sigma_k} - \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{R_{ij}\mu_{ij}} \qquad (6.3.1)$$

Where $\boldsymbol{\mu_{ij}}$ is already defined as distance desirability value of locating facility a_i adjacent to facility a_j. Table 6.2 shows the value of $\boldsymbol{\mu_{ij}}$ according to the distance between two facilities.

$\boldsymbol{\mu_{ij}}$	**Relationship condition**
1	$0 < Xp_{ij} + Yp_{ij} < (l_i + d_i)/4$
0.6	$(l_i + d_i)/4 < Xp_{ij} + Yp_{ij} < (l_i + d_i)/2$
0.3	$(l_i + d_i)/2 < Xp_{ij} + Yp_{ij} < 3*(l_i + d_i)/4$
0	$3*(l_i + d_i)/4 < Xp_{ij} + Yp_{ij} < (l_i + d_i)$

Table 6.2. The desirability values-based distance intervals

Subject to:

$$g(1) = \frac{l_i + l_j}{2} - (x_i - x_j + X_{max}(1 - z_{ij}^{x})) \qquad (6.3.2)$$

$$g(2) = \frac{d_i + d_j}{2} - (y_i - y_j + Y_{max}\left(1 - z_{ij}^{y}\right)) \qquad (6.3.3)$$

$$g(3) = x_i + \frac{l_i}{2} - L_r \ \ where\ s = U_i \qquad (6.3.4)$$

$$g(4) = y_i + \frac{d_i}{2} - D_r \ \ where\ s = U_i \qquad (6.3.5)$$

$$g(5) = \frac{l_i}{2} - x_i \ \ where\ s = U_i \qquad (6.3.6)$$

$$g(6) = \frac{d_i}{2} - y_i \ where\ s = U_i \qquad (6.3.7)$$

$$g(7) = \left[(x_i - x_j) - (1 - t_{ij})(X_{max} + Y_{max})\right] - Xp_{ij} \qquad (6.3.8)$$

$$g(8) = \left[(x_j - x_i) - (1 - t_{ij})(X_{max} + Y_{max})\right] - Xp_{ij} \qquad (6.3.9)$$

$$g(9) = \left[(y_i - y_c) - (y_c - y_j)\right] - Yp_{ij}\ where\ c = U_i \qquad (6.3.10)$$

$$g(10) = \left[(y_i - y_c) - (y_j - y_c)\right] - Yp_{ij}\ where\ c = U_i \qquad (6.3.11)$$

$$g(11) = \left[(y_c - y_i) - (y_c - y_j)\right] - Yp_{ij}\ where\ c = U_i \qquad (6.3.12)$$

$$g(12) = \left[(y_c - y_i) - (y_j - y_c)\right] - Yp_{ij}\ where\ c = U_i \qquad (6.3.13)$$

$$g(13) = (x_i - x_c) - Xc_i\ \forall\ i, c = 1 \qquad (6.3.14)$$

$$g(14) = (x_c - x_i) - Xc_i \ \forall\ i\ , \text{c}= 1 \tag{6.3.15}$$

$$g(15) = H_c \sum_{b=1}^{C} b\left(v_{ic} - v_{jc}\right) - Y_{c1c2} \ \forall\ i, j \neq i, c \neq 1 \tag{6.3.16}$$

$$g(16) = H_c \sum_{b=1}^{C} b\left(v_{jc} - v_{ic}\right) - Y_{c1c2} \ \forall\ i, j \neq i, c \neq 1 \tag{6.3.17}$$

$$g(17) = \left[Xc_i + \ Xc_j - \left(t_{ij} + o_{ij}\right)X_{max}\right] - Xp_{ij} \ \ \forall\ i, j \neq i, c \neq 1 \tag{6.3.18}$$

Constraints (6.3.2) and (6.3.3) insure the non-overlapping between two facilities, while (6.3.4)-(6.3.7) indicate that facilities have to be allocated within the appropriate row space defined by the corners of the department s (D_r, L_r) and $(0, 0)$. To calculate the distance between two facilities we use formulations (6.3.8)-(6.3.18). More details on the formulation can be found in §3.5.

6.3.2. Proposed algorithm for the OTLP

PSO is a robust stochastic optimization technique based on the movement and intelligence of swarms. It describes the collective behavior of decentralized, asynchronous, self-organized, natural, or artificial systems. As particles swarm over the solution space, they communicate with each other. They broadcast the fitness of their local best positions and the global best positions found by others to enhance the learning process. At each iteration (t), and based on their personal best Pbest_i as well as the global best $Gbest_i$, the particles adjust their velocities as follows:

$$V_i^{t+1} = \omega V_i^t + \beta_1 c_1\left(\text{Pbest}_i - x_i^t\right) + \beta_2 c_2\left(Gbest_i - x_i^t\right)$$

Where ω is the inertia weight to control the impact of the previous velocity on the current one, while c_1 and c_2 are the acceleration coefficients, which are also called respectively social and cognitive parameters. β_1 and $\beta_2$are two vectors of uniform random numbers between 0 and 1. After this update of the velocity, the position of each particle is updated using:

$$x_i^{t+1} = x_i^t + V_i^{t+1}$$

The general used PSO algorithm is described in Figure 6.6 and exposed in detail in the following points:

i) Initial layout generation

The PSO algorithm needs a number of initial solutions to initiate solution space exploration. We developed a heuristic for generating four different initial solutions based on a random permutation or based on frequencies of internal and external flows. Much research works proved that the quality of the initial solution has an influence on the quality of the final solution, and it tends to achieve early convergence to the best solution. Here we present a part of our heuristic generating initial layout solution based on internal and external flow frequency. Figure 6.7 shows an example of the arrangement of facilities in the available rectangular area with the fixed corridor obtained by this algorithm for one department.

Figure 6.6. Interaction overview diagram of the PSO algorithm

In this algorithm, for each department we start selecting the facility having the higher value of the traveling frequency with extern facilities (i.e. belonging to different departments). Once selected, we locate the facility 'i' as near as possible from the entry of the department and we select all facilities having dependency with it in the same department. Then, these selected facilities are located near the entry in decreasing order according to the traveling frequency with the facility 'i'. In each step, located facilities are dropped from the list of non-affected facilities (NAff). The algorithm is repeated until the NAff list is empty.

The second way to generate initial solution consists of randomly selecting the first facility 'i' on each iteration while dependent facilities are still located in a decreasing order according to the traveling frequency with the facility 'i'.

The third and fourth method for generating an initial solution resemble to the first and the second one respectively, just by selecting the lower value of the traveling frequency, and order dependent facilities in increasing order.

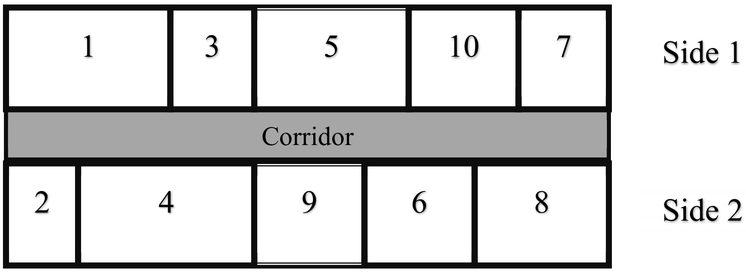


Figure 6.7. Example of initial solution

```
For each department:
    For each facility:
        Calculate the total relationship of each facility with all
        extern facilities (i.e. not belonging to the same department);
  End for
  Repeat:
        Select facility 'i' having max relationship with other
        facilities;
        Remove 'i' from list NAff (list of non-affected facilities);
        Place (i) as close as possible to the entry either in side 1
        or in side 2;
        Select facilities 'j' with highest R_ij = A(extremely necessary
        to 'i') and order them according to max relationship with 'i';
            For each 'j':
                Remove it from list NAff;
        Place it as close as possible from the entry either in side 1
        or in side 2;
      While (NAff is not empty);
 End for
```

Algorithm of the first method to generate initial solution for each department

ii) Particle Encoding

Most of PSO works applied discrete encoding to facility layout-based assignment problems (Kulturel-Konak & Konak [2011]) and (Rezazadeh et al. [2009]). Here a continuous encoding to OTLP is used where each department corresponds to a sequence of facilities, the side (row) they belong to, their dimensions and their locations in the x-y.

For solving OTLP with one department, we generate a population with four particles; each particle is encoded using a different initial solution. With N departments and P initial layout solutions for each department, we obtain a population with P^N particles, where a particle is an arrangement of N departments put in an order chosen from P distinguishable initial solutions. Table 6.3 shows an example of a particle with the initial solution in Figure 6.7 with one department.

Sequence of facilities	**Side**	**Dimension**	**Location**
1, 3, 5, 10, 7, 2, 4, 9, 6, 8	1, 1, 1, 1, 1, 2, 2, 2, 2, 2	$\{l_1, d_1\}$, $\{l_3, d_3\}$, $\{l_5, d_5\}$, $\{l_{10}, d_{10}\}$, $\{l_7, d_7\}$, $\{l_2, d_2\}$, $\{l_4, d_4\}$, $\{l_9, d_9\}$, $\{l_6, d_6\}$, $\{l_8, d_8\}$	$\{X_1, Y_1\}$, $\{X_3, Y_3\}$, $\{X_5, Y_5\}$, $\{X_{10}, Y_{10}\}$, $\{X_7, Y_7\}$, $\{X_2, Y_2\}$, $\{X_4, Y_4\}$, $\{X_9, Y_9\}$, $\{X_6, Y_6\}$, $\{X_8, Y_8\}$

Table 6.3. Representation of a particle with the initial solution in Figure 6.3.2

iii) Local search

To prevent particles from falling into local optimum traps and helping them to find better solution in less time, we embedded a local search algorithm into PSO to produce more satisfactory solutions and called it LS-PSO. The local search algorithm is applied at the end of an iteration of PSO in a way to improve the founded global best.

The global best is randomly modified by swapping two facilities based on both randomly generated indexes *k* and *l*. Swap operator generates a neighbor (solution) from the current candidate solution by making a slight change to it. After each swap, the solution is evaluated again and the new objective function value is compared with the global best one. If a better solution is found, the global best solution will be replaced and the local search is started again. These steps are repeated until the stopping criterion is met.

```
l, k = randomly generated indexes;
  X = best-solution of the iteration;
  Repeat:
        Swap (l, k, X);
        If FA(X) < FA(Gbest)
              Then Gbest <- X;
        Generate new l and k;
  While (stopping criterion is met);
```

Algorithm of the local search

iv) Constraint handling

The major challenge when solving FLP with PSO algorithm is that the FLP is a highly constrained engineering optimization problem whereas PSO does not have a mechanism to deal with constrained problems. In the literature, there are different ways of constraint handling in

PSO. He et al. (2004) used a constraint handling method called the fly-back mechanism to maintain a feasible population. They also extended the standard PSO algorithm to handle mixed variables using a simple scheme. Zahara and Hu (2008) proposed embedded constraint handling methods combined with the Nelder-Mead simplex search method. Their method included the gradient repair method and constraint fitness priority-based ranking method. Kulturel-Konak & Konak (2011) used a construction heuristic to generate acceptable layout structures and a penalty function to penalize solution infeasibilities because of department shapes and layout dimensions. In order to avoid premature convergence, Cagnina (2011) used a dynamic-objective constraint-handling method and an e-constraint constrained differential evolution.

In our case, to transform a constrained problem to an unconstrained one, a penalty function is used in which a penalty term in equation (6.3.19) is added to the objective function in equation (6.3.1). The aim is to decrease the degree of unfeasible solutions and favor the selection of feasible solutions. In equation (6.3.20), the violations of the constraints are not forbidden, but discouraged. When a constraint is violated (i.e. when $g_i(\vec{x}) > 0$) a big term is added to the objective function to push back the search towards to a feasible region.

$$p(\vec{x}) = \sum_{i=1}^{m} r_i * \max(0, g_i(\vec{x}))^2 \qquad (6.3.19)$$

$$Z = FA + p(\vec{x}) \qquad (6.3.20)$$

Where Z is the expanded objective function to be optimized, where r_i are positive constants called “penalty factors” and m is the number of violated constraints.

6.3.3. Numerical illustration

The proposed LS-PSO algorithm is validated on a set of generated data instances (one data). We used Windows 7 platform, Intel5® Core ™ i5-3380M CPU@ 2.90GHz and 8Go of RAM. The algorithm was implemented using JAVA as an Object-Oriented Programming language.

Figure 6.8 illustrates the modeled class diagram to develop our PSO in JAVA programming language. Seven classes are presented in this diagram with a non-exhaustive list of attributes and operations:

- Particle: In this class, all the functionality needed to run the PSO, i.e. defining the initial position and the initial velocity, save and update the current velocity and position of the particle, save and update the best and the global best solution, calculate the fitness function, etc.;

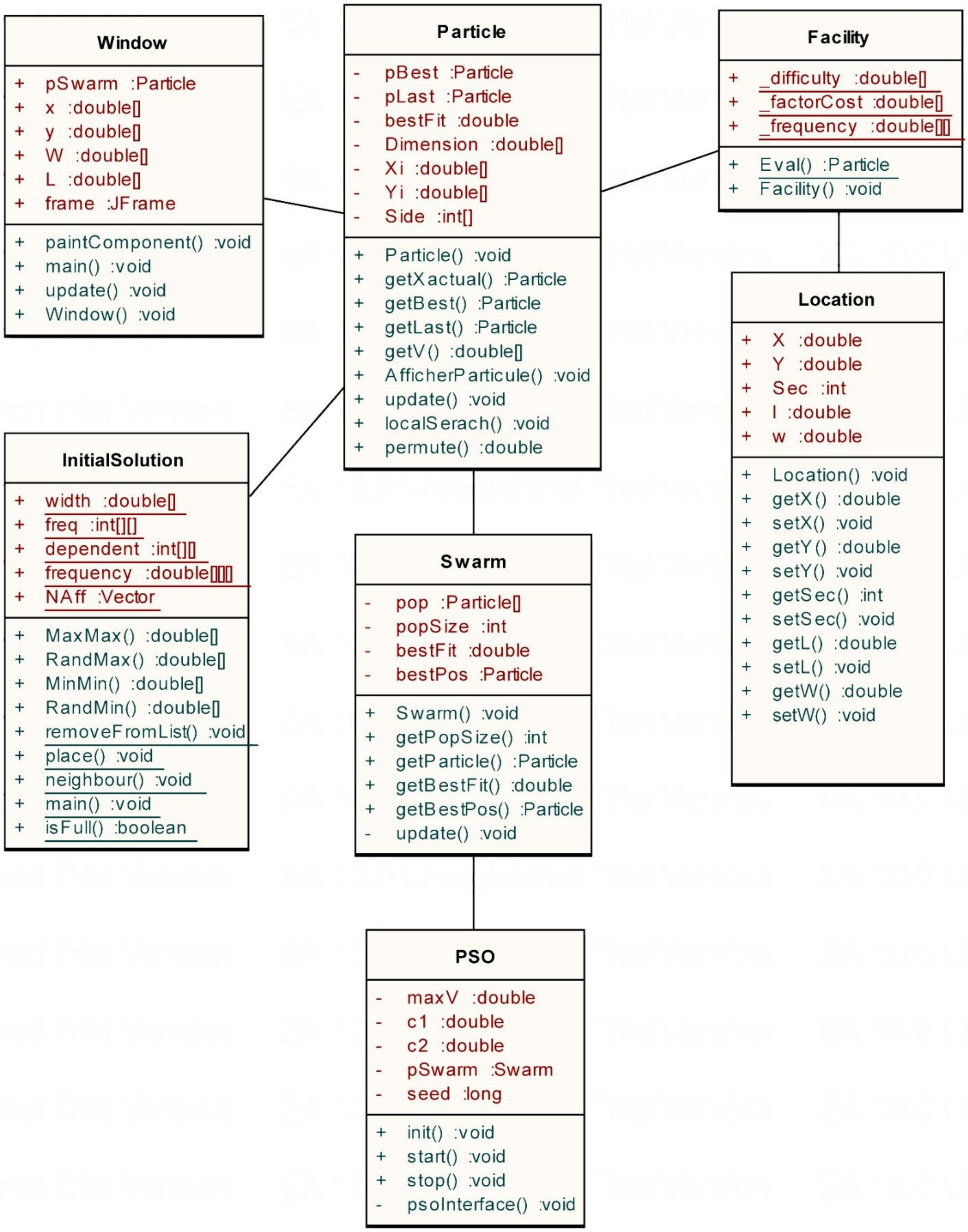


Figure 6.8. Class diagram of the developed PSO algorithm

- PSO: This class is responsible of initializing all PSO parameters, interface parameters, start, stop and destroy particles running;
- Swarm: This class implements the class 'Runnable', which should be implemented by

any class whose instances are intended to be executed by a thread. In addition, it extends the class 'Observable', which allows it to observe the particles' behavior and update the best solution once found;

- Facility: Here, all the parameters to solve the OTLP are defined, i.e. traveling frequency, traveling difficulty, department width and length, rows and corridor sizes and locations, etc.;
- Location: To identify the location of each facility in a particle we use this class;
- InitialSolution: this class is used to generate the four initial solutions for each department;
- Window: This class is the graphical interface in which the final OTLP solution is drawn.

An example is provided to illustrate the implementation of LS-PSO to the solution of OTLP with one department of twelve facilities. The heuristic generates four initial solutions to initialize the four particles ($P^N = 4^1$). Table 6.4 represents initial particles before running the LS-PSO search process, while Table 6.5 shows the new particles after one iteration. We can observe in Table 6.4 that the ***Gbest*** = 5408660 corresponds to third particle. In Table 6.5, particles 1, 2 and 4 improve their ***Pbest*** solutions, while the ***Gbest*** did not change.

n°	**Sequence of facilities**						**Fitness**	
1	12	7	3	2	9	4	5578260.0	*Pbest*[1]
	1	6	11	8	10	5		
2	5	9	10	2	8	4	7781180.0	*Pbest*[2]
	12	7	11	3	6	1		
3	9	12	7	5	4	3	**5408660.0**	*Pbest*[3] ***Gbest***
	6	10	1	11	8	2		
4	12	9	10	3	2	4	7334660.0	*Pbest*[4]
	5	8	11	7	6	1		

Table 6.4. Four particles representing the initial OTL solution

n°	**Sequence of facilities**						**Fitness**	
1	12	7	4	2	9	3	5468060.0	*Pbest*[1]
	1	10	11	8	6	5		
2	2	5	4	10	8	9	6099180.0	*Pbest*[2]
	3	7	11	6	12	1		
3	12	9	4	3	7	5	7941820.0	--
	8	10	11	1	2	6		
4	12	9	10	3	2	4	6568660.0	*Pbest*[4]
	5	8	11	7	6	1		

Table 6.5. Particles after one iteration

n°	Sequence of facilities						Fitness	
1	12 5	3 1	7 11	4 8	9 6	2 10	**5007511.0**	*Pbest*[1] ***Gbest***
2	9 1	5 7	2 11	8 6	4 3	10 12	6485980.0	--
3	7 1	12 8	5 11	4 2	9 6	3 10	531260.0	*Pbest*[3]
4	9 6	3 8	2 7	10 1	4 5	12 11	8615120.0	--

Table 6.6. Particles after 10 iterations

Running is stopped once the solution does not change anymore after 10 iterations. In this example, no change occurs after the tenth iteration and the retained solution is **Gbest** = 5007511 in less than three second. We solved each problem using different parameter values' combinations and we noted the best values, which permit early convergence. From these observations, we fixed the values of the following parameters as: $\omega = 0.8$; $c_1 = 2.0$; $c_2 = 2.0$; $r_i = 100$

We tested our LS-PSO for medium sized problems and compared the obtained solution with the optimal solution provided by the exact method. In terms of effectiveness, our LS-PSO reaches the same solution for all tested instance sizes. In terms of efficacy, PSO takes longer to solve these problems. LS-PSO is applied to the first data set of all instances sizes and comparison of the obtained results is reported in Table 6.7.

#	**Exact method**		**PSO**	
	Objective	Time (sec)	Objective	Time (sec)
ROW24	12763800	10	12763800	25
ROW28	14235423	25	14235750	57
ROW30	18523625	34	18523625	80
ROW32	20625688	52	20626105	91
ROW36	24498000	70	24498000	152
ROW40	29462400	120	29462400	213

Table 6.7. Exact method VS PSO for medium sized problems

Figure 6.9 gives an example of particle exploration in the research space, where the red point is the global best and the black points are the last position before moving to another position when converging to the global best. This is example of a problem with four departments and with four generated initial solutions $popSize = P^N = 4^4 = 256$ particles.

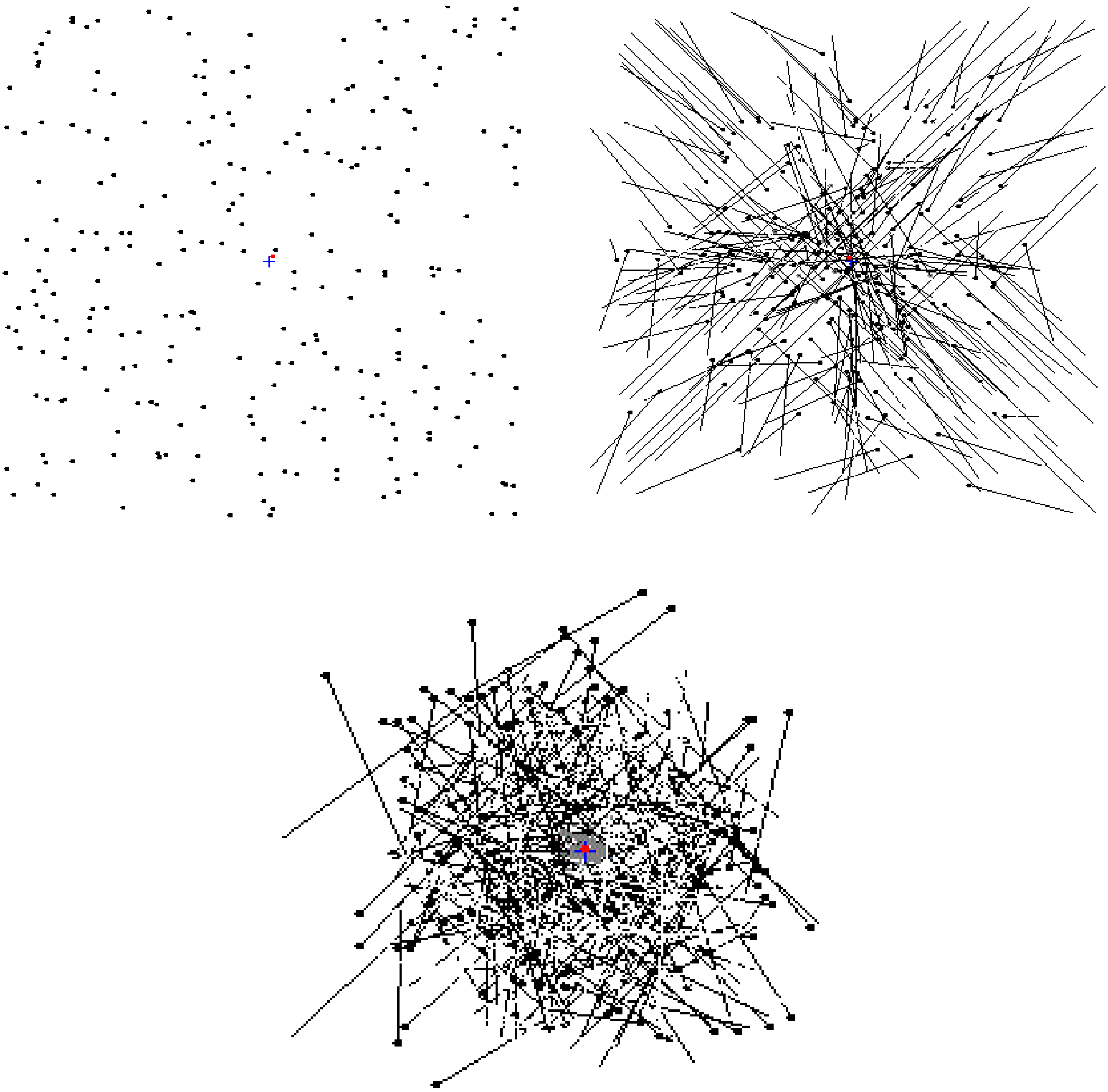


Figure 6.9. Particle behavior in the research space

For large sized OTLP, due to the reason that no past works have ever considered this assumption before, the obtained results were only compared to the exact method results. Each test problem was solved several times for the LS-PSO algorithm, and the best and average solutions were recorded.

The LS-PSO is a hybrid algorithm combining the advantages of PSO and local search methods in a way that the best solution found by the PSO in each iteration is improved by the local search. PSO is an effective algorithm to search and explore promising regions in the solution space in a distributed manner while maintaining excellent population diversity. The feature of the local search technique is to be effective when exploiting discovered promising regions of the solution space and using information related to the global best to improve itself.

To experiment the features of the LS-PSO, we solved the test problems using only the PSO without the local search, and using LS-PSO with the local search. Thus, the obtained results by these two algorithms are compared to each other and to the exact method, while the stopping criteria is set as the maximum CPU time allowed by Cplex to solve exact formulation before stopping.

Instance	Number of departments	Time (sec)	Exact method	PSO	LS-PSO
ROW60	5	18000	6646960	4446960	***4164790***
ROW72	6	15500	9107826	8452650	***7854250***
ROW84	7	14000	15963250	11200200	***10865260***
ROW96	8	14000	19875355	13659830	***11982440***
ROW108	9	12600	27325680	21358410	***18362410***

Table 6.8. Comparison of the exact method vs PSO vs LS-PSO

Table 6.8 presents the comparison made between exact method, PSO and LS-PSO, knowing that for exact method, we retain the last feasible solution with an average of the percentage gap between the lower bound and the upper bound equal to 40%. It is observable that the solutions provided by LS-PSO are more relevant and better than simple PSO. The reason for this is that PSO is effective as a global search method but does not have local search properties due to the random sampling of the solution space regions, while the LS significantly improves a solution through a rigorous search around the solution.

Our LS-PSO associates the capacity of exploration from the PSO and the capacity of exploitation from the LS for a more effective search. Using the two approaches' advantages, the LS-PSO continuously finds greater solutions.

6.3.4. Summary

This section presents the developed LS-PSO for solving multi-departments OTLP while formulating each department as a multi-rows layout problem. First, we introduce the mathematical model, the assumptions, the parameters, the constraints and the multi-objective function. Then the proposed PSO algorithm is exposed with the technique method to encode particles. The algorithm for generating initial solution as well as local search algorithm are also presented here.

Finally, the numerical experiments are illustrated while comparison has been made between LS-PSO, simple PSO and exact method. Obtained results proved efficiency of the proposed LS-PSO trough combining advantage of PSO and local search techniques.

This work is a first step in using metaheuristics for solving OTLP. Further works will explore other techniques for this problem to make comparisons and to combine their advantages in a hybrid method.

6.4. Conclusion

Operating theater layout problem and facility layout problems in general belong to the category of NP-hard problems. This makes exact methods inappropriate for solving large sized problems and requires the use of approximate methods to provide near optimal solution to studied problem as GA, TS, ACO, SA, PSO, etc. However, it is not easy to implement these methods for most of the real-world problems. Thus, the choice of the method to implement depends essentially on the studied problem.

The first section of this chapter introduces briefly the most-known approximate methods and highlights the advantages and disadvantages of each one. PSO was chosen to implement at first for a number of raisons, namely the simplicity of its implementation, its greater accuracy and its higher convergence speed.

PSO is methods issue from the swarm intelligence inspired by the metaphor of swarming behavior in nature by following the work of animal societies without leader in finding optimal values. The PSO algorithm is based on a set of autonomous particles arranged randomly and homogeneously in the search space. Each particle represents a potential solution to the problem, has a memory to remember its previous best position and communicates with other particles in its entourage. Particles have also a position and a velocity in the space search with which they will move evolving to an optimal or near-optimal solution in that space.

This chapter presents a PSO algorithm combined to a local search procedure to solve OTLP. Our PSO is using a heuristic to generate four different initial solutions for each department of the OT that will be used to encode each particle of the swarm. A good initial solution contributes to the early convergence to the best solution as we noted that the final solutions are dependent of the starting solutions and are usually quite different in configuration from the initial solution.

The problem was modeled as multi-departments layout problem, while each department is represented as a multi-rows layout. To experiment its efficiency, our PSO algorithm was tested to small medium and large sized problems for which it provides promising results.

To perform our formulation, further works will explore another method by relaxing the facilities dimension and the layout space constraints, solve it initially as a QAP, and then consider locally improving the solution by reintroducing the dimension and space constraints. This will allow us to make a comparative of all our developed methods in term of performance, CPU time and limits of each one.

6.5. Résumé

L'OTLP et le problème d'aménagement des installations en général appartiennent à la catégorie des problèmes NP-difficiles. Cela rend les méthodes exactes inappropriées pour résoudre les problèmes de grandes de tailles et nécessite l'utilisation de méthodes approchées pour fournir des solutions approximatives au problème étudié comme GA, TS, ACO, SA, PSO, etc. Cependant, il n'est pas facile à mettre en œuvre ces méthodes pour la plupart des problèmes du monde réel. Ainsi, le choix de l'approche à mettre en œuvre dépend essentiellement du problème étudié.

La première section de ce chapitre présente brièvement les méthodes approchées les plus connues et met en évidence les avantages et les inconvénients de chacune. Nous avons choisi de développer une PSO dans un premier temps pour un certain nombre de raisons, à savoir la simplicité de sa mise en œuvre, sa grande précision et sa grande vitesse de convergence.

La PSO est issue des méthodes de l'intelligence artificielle inspirée par la métaphore du comportement des groupes dans la nature en simulant le modèle de vie des sociétés d'animaux sans chef dans la recherche de valeurs optimales. L'algorithme de la PSO est basé sur un ensemble de particules autonomes disposées au hasard et de façon homogène dans l'espace de recherche. Chaque particule représente une solution potentielle au problème, a une mémoire de se souvenir de sa précédente meilleure position et communique avec d'autres particules dans son entourage. Les particules ont également une position et une vitesse dans l'espace de recherche avec lesquelles ils se déplaceront vers une solution optimale ou quasi-optimale dans cet espace.

Ce chapitre présente l'algorithme PSO combiné à une procédure de recherche locale pour résoudre l'OTLP. Notre PSO utilise une heuristique pour générer quatre solutions initiales différentes pour chaque département du bloc opératoire, qui seront par la suite utilisées pour coder chaque particule de l'essaim. Une bonne solution initiale contribue à la convergence rapide et facile pour une meilleure solution, comme nous avons pu constater que les solutions finales dépendent des solutions initiales et ne sont pas si différente dans la configuration de la solution initiale.

Le problème a été modélisé comme un problème de disposition en structure multi-département, tandis que chaque département est représenté comme une structure multi-rangée. Pour expérimenter son efficacité, notre algorithme PSO a été testé sur des problèmes de petites, moyennes et grandes tailles pour lesquelles il fournit des résultats prometteurs.

Pour améliorer notre formulation, d'autres travaux vont explorer une autre approche en relaxant les contraintes des dimensions des installations et des bornes de l'espace du bloc opératoire, résoudre d'abord le problème comme un QAP, et ensuite améliorer localement la solution en réintroduisant les contraintes de dimensions et d'espace. Cela nous permettra de faire un comparatif de toutes nos méthodes développées en termes de performance, des temps de calculs et les limites de chacune.

Chapter VII.

Conclusion and Future Research

7.1. Conclusion

The facility layout problem (FLP) is a common industrial problem of allocating facilities to specific designated locations. It seeks to determine the most efficient placement of facilities within a designated section of a plant, subject to some constraints imposed by the physical departmental areas, system-operating requirements and the desire of the decision-makers. The FLP has practical and theoretical importance. It has been widely used by researchers to design floor layouts of offices in buildings, airports, warehouses, etc. but it still less addressed in healthcare context. This work come to fill the gap in the literature by proposing a solution to FLP applied to OT layout.

For this, the first chapter starts with an introduction in which the motivations of this work are presented. Then a brief presentation of different aspects of FLP is provided to initiate the reader with some of the used notion in this dissertation. Finally, the objectives of this work are listed to highlight the major contributions presented in this dissertation.

The second chapter presents a comprehensive literature review of the static and dynamic FLP based on a large number of literature references. It was observed that research and application works on the facility layout problem continued in several domains including more complex and realistic manufacturing and service industries.

Various approaches were employed to solve different facility layout variants with specific objective functions and constraints to meet the characteristics of the studied system. The presented works have been classed according to environment evolution, either the static or the dynamic FLP while in each category articles have been sorted on the basis of the proposed approach namely, exact methods, heuristics or meta-heuristics. A literature review of the works dealing with FLP in healthcare system is then provided to show the lack of such research in this domain and to highlight the motivation of our work to apply FLP to OT layout (OTL) design.

In order to be relevant and rigorous in addressing the need of modern flexible and efficient health care systems, four models were developed in Chapter 3 for two-dimensional OT layout under Static formulation namely, simple OTL; multi-section OTL; multi-floor OTL and multi-row OTL. The motivation of these formulations came from observation of realistic case of OTL, the understanding of their operating modes and the comprehension of hospitals' needs.

For each formulation, a MIP model is developed to determine the optimal location and orientation of each facility in the final OTL while minimizing the total traveling cost taking into account four different actors of the OTL (i.e. doctors, patients, medical and non-medical staff) and maximizing the desirable closeness rating of facilities. The applicability of the proposed model was demonstrated on several instances to provide optimal solutions for reasonable real sized instances. This chapter shows the evolution of the exact method to solve Static OTL both in terms of sizes of instances and in terms of computational results. Several assumptions were taking into account based on real OTL specifications and properties, while respecting international standards that regulate the healthcare facilities design.

The size of instance, for which an optimal solution is provided, depends on the assumed formulation. We are able to model and solve instances up to forty facilities with our exact formulation. These results represent an advance in this research field while works in the literature using MIP were not able to find exact solutions for instances of thirty facilities.

The fourth chapter presents also three MIP formulations to solve Dynamic OTL (DOTL) using two different approaches. The FFLP consists of finding only one single robust OT layout for the whole planning horizon. In contrast, the VFLP seeks to generate a possibly different OT layout for each period in the planning horizon. In this chapter, we tested the performance of the proposed formulations to solve DOTL under several assumptions and constraints for different sized instances while considering different planning horizon segmentations.

Due to unavailable realistic values for the flows of different actors, their costs, building and rearrangement costs, we generate a set of quasi-real data instances based on real life observations and inquiry in some hospitals of our contacts. The computational results were optimal with FFLP in a reasonable time for the three MIP formulations, while the designed layouts were satisfying international standards in term of safety and hygiene. For VFLP, we did not attain optimality for bigger instances in multi-sections and multi-floors, but the final OT layouts show that we obtained the same layout in some cases. The FFLP approach gave good performances equivalent to the static formulation's results, while VFLP was not so far from the static formulation's performance due to the complexity of this approach in terms of variables and constraints number.

Optimal solutions were provided for instances up to forty facilities. Exact solutions for such sizes did not exist in the literature since dynamic FLP was generally solved with approximate methods or for small sized instances as reviewed in Section 2.2.

Operating theater layout problem and facility layout problems in general belong to the category of NP-hard problems. This makes exact methods inappropriate for solving large sized instances for which the use of approximate methods are proposed to provide near optimal solutions. In Chapter 5, a Multi-Agent System (MAS) architecture is presented to solve multi-departments OTLP with static and dynamic environments based on multi-section formulation in Sections 3.3 and 4.2. The proposed method was tested on large-sized OTL for which exact methods were unable to obtain solutions for real cases of OTL that exists in big hospitals. Our MAS provides an efficient approach for solving large-sized OTL using different communication protocols (i.e. cooperation, collaboration and negotiation). In terms of instances' size, we solved problems with 162 facilities for the static OTL and 88 facilities for dynamic OTL but it is not necessarily limited to these numbers. In terms of processing time, final solutions were obtained efficiently in a reasonable and reduced computational time effort as compared to those needed to obtain by exact methods.

The sixth chapter presents a Particle Swarm Optimization (PSO) algorithm combined with a local search procedure to solve OTL. The developed PSO uses a heuristic to generate four different initial solutions for each department of the OTL that will be used to encode each particle of the swarm. A good initial solution contributes to the early convergence to the best solution. We found that the final solutions are dependent on the quality of the starting solutions and are usually quite different from the configuration of the initial solutions. The problem was modeled as multi-departments layout problem, while each department is represented as a multi-row layout. To explore the efficiency of our PSO algorithm, it was tested small to medium and large sized instances for which it provides very good and promising results.

We can summarize the major contributions of our work in the following three points:

- The application of MIP to OTL: This work considers three different formulations (the multi-section, the multi-floor and the multi-row layouts) in two different environment types (static and dynamic) while optimizing two different objective functions (minimize the total traveling cost and maximize the total adjacency rate). The combination of these different components give rise to nine MIP models to solve the OTL for which optimal solutions were provided to instances up to forty facilities.
- The use of MAS to solve FLP: In literature, only one work [Tarkesh et al., (2009)] applied the MAS to solve small-sized instances. Our work presents the first application of MAS to both the static and dynamic FLP for large-sized instances using a novel algorithm using three steps to solve OTL.
- The use of PSO under continuous layout representation to solve multi-row FLP: Since the PSO is generally used to solve assignment problems or discrete FLP, our actual formulation is among the few research works that considered the continuous type. This leads us to conceive a novel encoding technique and the appropriate heuristics to generate initial solutions and to perform the local search procedure. Another novelty is related to the application of PSO to a multi-row layout problem,

which was not addressed before in the literature. To the best of our knowledge, PSO works formulated the FLP as a single row or in the best of scenarios, as a double-row problem.

This work is an advanced contribution; as OT layouts are usually planned manually without taking into account the optimization aspects and international standards. The developed models can be used as a decision support tool to planners seeking optimal OT layout design since the considered assumptions and the obtained results were approved by a senior manager of OT in Roanne's hospital.

7.2. Further works

The actual work could further benefit by considering other important aspects. These perspectives can be classified into two categories:

7.2.1. Assumptions and formulations:

In terms of assumptions, this work can be extended by taking into account other important bricks of the OTL as to implant ventilation system, to include constraints related to emergency exit and waste disposal in the new model and to consider all categories of actors operating in the OTL.

In terms of formulation, several extensions can be proposed. In dynamic environment, it may happen that a facility (e.g. operating room, waiting space, etc.) needs an additional space in the next period of a planning horizon. Thus, one of the extensions could be the use of facilities with varying sizes from one period to another.

To calculate the traveled distance from a facility to another, it is not always rigorous to use the centroid-to-centroid measure. To improve this point, we propose to implement models that support the distance based on Input/ Output points. The use of flexible bay is also proposed to replace the use of corridors with variable coordinates, which makes the problem more complex as a way to solve larger instances.

Until now, we used standards facilities with rectangular shapes to represent OTL rooms. This rectangular assumption may not always be the case in all hospitals where facilities could take irregular shapes. Thus, further works would consider other forms in the problem formulation. Furthermore, the irregular shapes will be considered for the whole layout space to treat the real case of OTL where architects could face construction site with non-rectangular boundary.

7.2.2. Methods and tools:

With regards to methods for solving OTL, further works will consider improving further the developed algorithms (i.e. MAS and PSO) to cover the new assumptions and formulations cited above.

For the developed MAS, further works will focus firstly on developing other communication protocols to compare the efficiency of the used ones and secondly on increasing the performance of each agent to solve larger-sized instances by combining MAS either with another intelligent method or with other meta-heuristics.

For the developed PSO, we also plan to explore another approach by relaxing the facilities dimension and the layout space constraints and solve it initially as a QAP and then consider improving the QAP solution by reintroducing the added constraints on dimension and space. This will allow us to make a comparative of all our developed methods in term of quality of solutions, and CPU time and limitations of each approach.

The fusion of MAS and PSO is also a perspective to consider for evaluating the performance of the hybrid methods. To put to the proof the effectiveness of developed PSO compared to existent approximate methods, we intend to implement the OTL using other meta-heuristics such as genetic algorithm, tabu search, simulated annealing among others.

Furthermore, one of our major perspective is to commercialize this application to an architect's offices or directly to hospitals. For this, an interactive graphical interface is needed to make the tool more attractive and easy to handle. This product will not be long in coming since we are working seriously on the GUI and putting the necessary component to its well performing.

Chapitre VII.

Conclusion et perspectives

7.1. Conclusion

Le problème d'agencement des installations FLP est un problème industriel commun d'affectation des installations à des emplacements désignés. Il vise à déterminer l'emplacement le plus efficace des installations au sein d'une section désignée d'un plan, sous réserve de certaines contraintes imposées par le site de construction, les exigences d'exploitation du système et la volonté des décideurs. Le FLP a une importance pratique et théorique. Il a été largement utilisé par les chercheurs pour concevoir des plans d'étage des locaux dans des bâtiments, des aéroports, des entrepôts, etc., mais il est encore moins abordé dans le contexte des établissements de santé. Ce travail vient pour combler ce vide dans la littérature, en proposant une solution au FLP appliqué au bloc opératoire.

Pour cela, le premier chapitre commence par une introduction dans laquelle les motivations de ce travail sont présentées. Ensuite, une brève présentation des différents aspects du FLP est donnée pour initier le lecteur avec une partie des notions utilisées dans cette thèse. Enfin, les objectifs de ce travail sont énumérés pour mettre en évidence les principales contributions.

Le deuxième chapitre présente un état de l'art du FLP statique et dynamique basé sur un grand nombre de références bibliographiques. Il a été observé que la recherche et l'application sur le FLP avançaient dans plusieurs domaines incluant des problèmes d'industries manufacturières plus complexes et réalistes.

Différentes approches ont été utilisées pour résoudre différentes variantes du FLP avec des fonctions objectifs et des contraintes spécifiques pour répondre aux caractéristiques du système étudié. Les œuvres présentées ont été classées en fonction de l'évolution de l'environnement, soit statique ou dynamique, tandis que dans chaque catégorie, les articles ont été triés sur la base de l'approche utilisée, à savoir des méthodes exactes, heuristiques ou métaheuristiques. Un état de l'art des travaux traitant de FLP dans les établissements de soins et de santé est ensuite présenté pour montrer l'absence de ce type de recherche dans ce domaine et de mettre en évidence la motivation de notre travail à appliquer le FLP aux blocs opératoires (OTLP).

Pour être pertinent et rigoureux pour répondre aux besoins des systèmes de soins et de santé modernes, quatre modèles ont été développés pour le OTLP sous formulation statique : a. simple OTLP ; b. OTLP à plusieurs sections ; c. OTLP à plusieurs étages et OTLP à plusieurs rangées. La motivation de ces formulations est venue de l'observation de quelques blocs opératoires, la compréhension de leurs modes de fonctionnements et la compréhension des besoins des hôpitaux.

Pour chaque formulation, un modèle MIP est conçu pour déterminer l'emplacement et l'orientation optimale pour chaque installation dans l'OTL final tout en minimisant le coût total de déplacement en tenant en compte les différents acteurs du bloc opératoire (chirurgiens, patients, équipe médicale et non-médicale) et en maximisant le taux de proximité souhaitable. L'applicabilité du modèle proposé a été démontrée sur plusieurs cas pour fournir des solutions optimales pour les problèmes réels de tailles raisonnables.

Ce chapitre montre l'évolution de la méthode exacte pour résoudre l'OTLP statique tant en termes de tailles des instances qu'en termes de résultats de calculs. Plusieurs hypothèses ont été prises en compte sur la base de spécifications réelles et de propriétés du bloc opératoire, tout en respectant les normes internationales qui régissent la conception des établissements de soins et de santé.

La taille des instances, pour lesquelles une solution optimale est trouvée, dépend de la formulation étudiée. Nous sommes en mesure de modéliser et de résoudre des instances jusqu'à quarante installations avec notre formulation exacte. Ces résultats représentent une avancée dans ce domaine de recherche tandis que les travaux dans la littérature en utilisant MIP ne sont pas en mesure de trouver des solutions exactes pour les instances de trente installations.

Le quatrième chapitre présente trois formulations MIP pour résoudre l'OTLP dynamique en utilisant deux approches différentes. Le FFLP (disposition avec installations fixes) consiste à trouver en une seule décision unique un agencement robuste du bloc opératoire pour tout l'horizon de planification. En revanche, le VFLP (disposition avec installations amovibles) cherche à générer un agencement du bloc opératoire pour chaque période de l'horizon de planification. Dans ce chapitre, nous avons testé les performances des formulations proposées pour résoudre l'OTLP dynamique sous plusieurs hypothèses et contraintes pour des problèmes de tailles différentes et des segmentations différentes de l'horizon de planification.

En raison de non disponibilité des valeurs réalistes des flux de différents acteurs, leurs coûts, des coûts de construction et de réarrangement, nous avons généré un ensemble de données pour des instances de taille différentes fondées sur des observations réelles dans certains hôpitaux. Les résultats de calculs étaient optimaux avec le FFLP dans un délai raisonnable pour les trois formulations MIP, tandis que les dispositions conçues étaient satisfaisantes aux normes internationales en termes de sécurité et d'hygiène. Pour VFLP nous n'avons pas atteint l'optimalité pour les instances de grandes tailles dans le cas de l'OTLP en multi-section et en multi-étage, mais les dispositions finales des blocs opératoires montrent que, même pour l'écart restant, on obtient la même disposition dans certains cas du FFLP. L'approche du FFLP a donné de bonnes performances qui étaient équivalentes aux résultats de formulation statique, tandis que le VFLP n'était pas si loin de la performance de la formulation statique due à la complexité de cette approche en termes du nombre de variables et de contraintes.

Des solutions optimales ont été trouvées pour les instances jusqu'à quarante installations. Des solutions exactes pour ces tailles n'existent dans la littérature du moment où le FLP dynamique a été généralement résolu avec des méthodes approchées ou pour des instances de petites de tailles tel que présenté dans la section 2.2.

L'OTLP et le problème d'aménagement des installations en général appartiennent à la catégorie des problèmes NP-difficiles. Cela rend les méthodes exactes inappropriées pour résoudre les problèmes de grandes de tailles et nécessite l'utilisation de méthodes approchées pour fournir des solutions approximatives aux problèmes étudiés. Une architecture de système multi-agent a été présentée dans le cinquième chapitre pour résoudre le problème de disposition du bloc opératoire pour une structure à plusieurs départements sous un environnement statique et dynamique basé sur la formulation de structures à plusieurs sections vue dans §3.3 et §4.2. La méthode proposée a été testée sur des OTLP de grandes tailles pour lesquels les méthodes exactes étaient incapables de donner des solutions ; et qui représentent un cas réel de bloc opératoire dans les grands hôpitaux.

Le système multi-agent offre une capacité très performante pour résoudre des OTLP de grandes tailles en utilisant des protocoles de communication différents (à savoir la coopération, de collaboration et de négociation). En termes de tailles d'instances, nous avons résolu des problèmes avec 162 installations pour l'OTLP statique et 88 installations pour l'OTLP dynamique mais nous ne sommes pas nécessairement limités à ces nombres. En termes de temps de traitement, des solutions ont été obtenues dans un temps réduit par rapport à celles obtenues avec des méthodes exactes.

Dans le sixième chapitre, il a été présenté l'algorithme PSO combiné à une procédure de recherche locale pour résoudre l'OTLP. Notre PSO utilise une heuristique pour générer quatre solutions initiales différentes pour chaque département du bloc opératoires, qui seront par la suite utilisées pour coder chaque particule de l'essaim. Une bonne solution initiale contribue à la convergence rapide et facile pour une meilleure solution, comme nous avons pu constater que les solutions finales dépendent des solutions initiales et ne sont pas si différentes dans la configuration de la solution initiale.

Le problème a été modélisé comme un problème de disposition en structure multi-département, tandis que chaque département est représenté comme une structure multi-rangée. Pour expérimenter son efficacité, notre algorithme PSO a été testé sur des problèmes de petites, moyennes et grandes tailles pour lesquelles il fournit des résultats prometteurs.

Nous pouvons résumer les principales contributions de notre travail dans les trois points suivants :

- L'application du MIP au OTLP : Ce travail considère trois formulations différentes (multi-section, multi-étage et multi-rangée) dans deux types d'environnement différents (statique et dynamique), tout en optimisant des fonctions objectifs différentes (minimiser le coût totale de déplacement de maximiser le taux de proximité et minimiser les coûts de réarrangement). La combinaison de ces différentes composantes donne lieu à neuf modèles MIP pour résoudre l'OTLP pour des solutions optimales qui ont été trouvées à des instances avec des tailles allant jusqu'à quarante installations.
- L'utilisation de MAS pour résoudre le FLP : Dans la littérature, un seul travail [Tarkesh et al, (2009)] a appliqué le MAS pour résoudre de petites instances. Notre travail présente la première application du MAS pour à la fois le FLP statique et dynamique pour de grandes instances à l'aide d'un algorithme basé sur une architecture à trois étapes pour résoudre OTLP.
- L'utilisation de PSO sous une représentation continue de l'espace pour résoudre le FLP en multi-rangée : Puisque la PSO est généralement utilisée pour résoudre les problèmes d'affectation ou FLP sous une représentation discrète, notre formulation réelle est parmi les rares travaux de recherche qui ont étudié le type continu. Cela nous a amené à concevoir une nouvelle technique de codage et des heuristiques appropriées pour générer des solutions initiales et pour effectuer la procédure de recherche locale. Une autre nouveauté est liée à l'application des PSO à un problème de disposition en structure multi-rangée, qui n'a pas été abordé auparavant dans la littérature. A notre connaissance, les travaux ayant utilisés la PSO ont formulé le FLP comme une structure à une seule rangée ou dans le meilleur des scénarios, comme une structure à deux rangées.

Ce travail représente est une avancée cruciale dans le domaine de l'architecture et de l'industrie puisque la conception habituellement des blocs opératoires est planifiée manuellement sans prendre en compte les aspects d'optimisation. Les modèles développés peuvent être utilisés comme un outil d'aide à la décision pour les planificateurs pour optimiser la conception des blocs opératoires puisque les hypothèses considérées et les résultats obtenus ont été approuvés par un cadre supérieur du bloc opératoire à l'hôpital de Roanne.

7.2. perspectives

Le travail actuel pourrait être amélioré en considérant d'autres aspects importants. Ces perspectives peuvent être classées en deux catégories :

7.2.1. Hypothèses et formulations :

En termes d'hypothèses, ce travail peut être étendu en prenant en compte d'autres briques importantes du bloc opératoire comme intégrer le système de ventilation, inclure les contraintes liées aux sorties d'urgence et de l'élimination des déchets et envisager toutes les catégories d'acteurs opérant dans le bloc opératoire.

En termes de formulation, plusieurs extensions peuvent être proposées. Dans un environnement dynamique, il peut arriver qu'une installation (par exemple, la salle d'opération, l'espace d'attente, etc.) ait besoin d'un espace supplémentaire dans la prochaine période d'un horizon de planification. Ainsi, l'une des extensions pourrait être l'utilisation des installations avec des tailles variable d'une période à l'autre.

Pour calculer la distance parcourue à partir d'une salle à une autre, il n’est pas toujours rigoureux d’utiliser la mesure barycentre à barycentre. Pour améliorer ce point, nous proposons de mettre en œuvre des modèles qui prennent en charge la distance basée sur les points d'entrée / sortie. L'utilisation des cloisons est également proposée pour remplacer l'utilisation des couloirs fixes, ce qui rendra le problème plus complexe comme un moyen de résoudre des instances de plus grande taille.

Jusqu'à présent, nous avons utilisé les installations standards avec des formes rectangulaires pour représenter les salles du bloc opératoires. Cette hypothèse peut ne pas être toujours le cas dans tous les hôpitaux où les installations pourraient prendre des formes irrégulières. Ainsi, les prochaines pistes seraient d’envisager d'autres formes dans la formulation du problème. En outre, les formes irrégulières seront considérées aussi pour le site de construction pour traiter des blocs opératoires réalistes puisque les architectes peuvent faire face à des chantiers de construction avec des fromes non-rectangulaire.

7.2.2. Méthodes et outils :

En ce qui concerne les méthodes pour résoudre l’OTLP, d'autres travaux seront en outre envisager pour améliorer les algorithmes développés (MAS et PSO) pour couvrir les nouvelles hypothèses et formulations citées ci-dessus.

Pour le MAS développés, d'autres travaux porteront tout d'abord sur le développement d'autres protocoles de communication pour comparer l'efficacité de ceux utilisés et d'autre part sur l'augmentation de la performance de chaque agent pour résoudre des cas de plus grande taille en combinant MAS soit avec une autre méthode intelligente ou avec d'autres métaheuristiques.

Pour la PSO développées, nous prévoyons également d'explorer une autre approche en relaxant les contraintes de dimension des installations et des bornes de l'espace du bloc opératoire, résoudre d'abord le problème comme un QAP, et ensuite améliorer localement la solution en réintroduisant les contraintes de dimensions et d'espace. Cela nous permettra de faire un comparatif de toutes nos méthodes développées en termes de performance, de temps de calculs et les limites de chacune.

La fusion de la MAS et PSO est également une piste à considérer pour évaluer la performance des méthodes hybrides. Pour mettre à l'épreuve l'efficacité de la PSO développée par rapport aux méthodes approchées existantes, nous avons l'intention de résoudre l'OTLP en utilisant d'autres métaheuristiques telles que l'algorithme génétique, recherche tabou, recuit simulé entre autres.

En outre, l'un de nos perspectives le plus important est de commercialiser cette application pour les bureaux d'architecte ou directement aux hôpitaux. Pour cela, une interface graphique interactive est nécessaire pour rendre l'outil plus attrayant et facile à manipuler. Ce produit ne tardera pas à arriver puisque nous travaillons sérieusement sur l'interface graphique pour mettre le composant nécessaire à son fonctionnement.

References

Al-Hakim, L. A. (1992). A modified-procedure for converting a dual graph to a block layout. *The International Journal Of Production Research*, *30*(10), 2467-2476.

Anstreicher, K.M., N.W. Brixius, J. Linderoth, and J.P. Goux (2002). Solving Large Quadratic Assignment Problems on Computational Grids. Mathematical Programming, Series B 91, 563-588.

Arnolds, I. V., & Nickel, S. (2013). Multi-period layout planning for hospital wards. Socio-Economic Planning Sciences, 47(3), 220-237.

Arnolds, I., Nickel, S., Shashaani, S., & Wernz, C. (2012, December). Using simulation in hospital layout planning. In Proceedings of the Winter Simulation Conference (p. 395). Winter Simulation Conference.

Asl, A. D., & Wong, K. Y. (2015). Solving unequal-area static and dynamic facility layout problems using modified particle swarm optimization. Journal of Intelligent Manufacturing, 1-20.

Assem, M., Ouda, B., & Abdel Wahed, M., (2012). Improving Operating Theatre Design Using Facilities Layout Planning. Cairo International Biomedical Engineering Conference (CIBEC), Cairo, Egypt, December 20-21, 2012

Azadeh, A., Jebreili, S., Chang, E., Saberi, M., & Hussain, O. K. An integrated fuzzy algorithm approach to factory floor design incorporating environmental quality and health impact. International Journal of System Assurance Engineering and Management, 1-12.

Azimi, P., & Charmchi, H. M. (2011). A new heuristic algorithm for the dynamic facility layout problem with budget constraint. In Proceedings of International Conference on Modelling and Simulation (ICMS 2011).

Balakrishnan, J., & Cheng, C. H. (1998). Dynamic layout algorithms: a state-of-the-art survey. Omega, 26(4), 507-521.

Balakrishnan, J., & Cheng, C. H. (2009). The dynamic plant layout problem: Incorporating rolling horizons and forecast uncertainty. Omega, 37(1), 165-177.

Balakrishnan, J., Cheng, C. H., & Conway, D. G. (2000). An improved pair-wise exchange heuristic for the dynamic plant layout problem. International Journal of Production Research, 38(13), 3067-3077.

Balakrishnan, J., Jacobs, F. R., & Venkataramanan, M. A. (1992). Solutions for the constrained dynamic facility layout problem. European Journal of Operational Research, 57(2), 280-286.

Barrett, A. (2008). Optimization of facility design and workflow process at the phlebotomy clinic of Toronto general hospital. Toronto: University of Toronto

Barsegar, J. (2011). A Survey of Metaheuristics for Facility Layout Problems. University of Rhode Island.

Baykasoglu, A., Dereli, T., & Sabuncu, I. (2006). An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. Omega, 34(4), 385-396.

Bazaraa MS, Sherali HD (1980). Bender's partitioning scheme applied to a new formulation of the quadratic assignment problem. Oaval Res Logistics Quart 27(1):29–41

Bazaraa, M. S., & Sherali, H. D. (1982). On the use of exact and heuristic cutting plane methods for the quadratic assignment problem. Journal of the Operational Research Society, 991-1003.

Bozer, Y. A., & Wang, C. T. (2012). A graph-pair representation and MIP-model-based heuristic for the unequal-area facility layout problem. European Journal of Operational Research, 218(2), 382-391.

Brixius, N.W., Anstreicher, K.M. (2001). Solving quadratic assignment problems using convex quadratic programming relaxations. Optimization Methods and Software 16, 49–68.

Burkard, R. E., & Bönniger, T. (1983). A heuristic for quadratic Boolean programs with applications to quadratic assignment problems. European Journal of Operational Research, 13(4), 374-386.

Butler, T. W., K. R. Karwan, J. R. Sweigart, and G. R. Reeves (1992). An Integrative Model-Based Approach to Hospital Layout. IIE Transactions 24(2), 144–152.

Carrie, A. S., Moore, J. M., Roczniak, M., & Seppänen, J. J. (1978). Graph theory and computer aided facilities design. Omega, 6(4), 353-361.

Chen*, C. W., & Sha, D. Y. (2005). Heuristic approach for solving the multi-objective facility layout problem. International Journal of Production Research, 43(21), 4493-4507.

Chen, G. Y. H. (2013). A new data structure of solution representation in hybrid ant colony optimization for large dynamic facility layout problems. International Journal of Production Economics, 142(2), 362-371.

Chen, M. (1998). A mathematical programming model for system reconfiguration in a dynamic cellular manufacturing environment. Annals of Operations Research, 77, 109-128.

Chraibi, A., Kharraja, S., Osman, I. H., & Elbeqqali, O. (2013, October). A mixed integer programming formulation for solving operating theatre layout problem: A multi-goal approach. In Industrial Engineering and Systems Management (IESM), Proceedings of 2013 International Conference on (pp. 1-10). IEEE.

Chraibi, A., Kharraja, S., Osman, I. H., & Elbeqqali, O. (2014). A Multi-objective Mixed-Integer Programming model for a multi-section Operating Theatre Facility Layout. In International Conference on Operations Research and Enterprise Systems (ICORES), Proceedings of 2014 International Conference.

Chraibi, A., Kharraja, S., Osman, I. H., & Elbeqqali, O. (2014, November). Solving operating theater facility layout problem using a Multi-Agent system. In Control, Decision and Information Technologies (CoDIT), 2014 International Conference on (pp. 207-212). IEEE.

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.

Chraibi, A., Kharraja, S., Osman, I. H., & Elbeqqali, O. (2015, October). Optimization of a Dynamic Operating Theatre Layout Problem. In Industrial Engineering and Systems Management (IESM), Proceedings of 2015 International Conference. IEEE.

Christofides, N., & Benavent, E. (1989). An exact algorithm for the quadratic assignment problem on a tree. Operations Research, 37(5), 760-768.

DARES : Conditions, Organisation du travail et Nouvelles Technologies en 2013, Dossiers statistiques du travail et de l'emploi.

Daskin, M.S., Dean, L.K., 2004. 3. Location of health care facilities. In: Brandeau, M.L., Sainfort, F., Pierskalla, W.P. (Eds.), Operations Research and Health Care. Kluwer, pp. 43–76.

Dickey, J.W., Hopkins, J.W., 1972. Campus building arrangement using Topaz. Transportation Research 6, 59–68.

Duman, E., Ilhan, O., in press. The quadratic assignment problem in the context of the printed circuit board assembly process. Computers and Operations Research

Dunker, T., Radons, G., & Westkämper, E. (2005). Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. European Journal of Operational Research, 165(1), 55-69.

Elshafei AN. Hospital layout as a quadratic assignment problem. Operational Research Quarterly 1977;28(1):167e79.

Engelbrecht, A. P. (2013, September). Particle Swarm Optimization: Global Best or Local Best?. In Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on (pp. 124-135). IEEE.

Erel, E., Ghosh, J. B., & Simon, J. T. (2003). New heuristic for the dynamic layout problem. Journal of the Operational Research Society, 54(12), 1275-1282.

Erel, E., Ghosh, J. B., & Simon, J. T. (2005). New heuristic for the dynamic layout problem. Journal of the Operational Research Society, 56(8), 1001-1001.

Ferber, J. Les systèmes multi-agents. Vers une intelligence collective. InterEditions, Paris, 1995

Feyzollahi, M., Shokouhi, A., Modarres Yazdi, M., & Tarokh, M. (2009). Designing a model for optimal hospital unit layout. Pajoohandeh Journal, 14(4), 191–198.

Foulds, L. R., & Robinson, D. F. (1976). A strategy for solving the plant layout problem. Operational Research Quarterly, 845-855.

Foulds, L. R., & Robinson, D. F. (1978). Graph theoretic heuristics for the plant layout problem. The International Journal of Production Research, 16(1), 27-37.

García-Hernández, L., Pierreval, H., Salas-Morera, L., & Arauzo-Azofra, A. (2013). Handling qualitative aspects in Unequal Area Facility Layout Problem: An Interactive Genetic Algorithm. Applied Soft Computing, 13(4), 1718-1727.

Gibson, I. W. (2007, December). An approach to hospital planning and design using discrete event simulation. In Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come (pp. 1501-1509). IEEE Press.

Gilmore, P. C. (1962). Optimal and suboptimal algorithms for the quadratic assignment problem. Journal of the Society for Industrial & Applied Mathematics, 10(2), 305-313.

Goetschalckx, M. (1992). SPIRAL: An efficient and interactive adjacency graph heuristic for rapid prototyping of facilities design. European Journal of Operational Research, 63, 304-321.

Hahn, P., Grant, T., Hall, N. (1998). A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method. European Journal of Operational Research 108, 629–640.

Hahn, P., J. MacGregor Smith, and Y.-R. Zhu (2010). The Multi-Story Space Assignment Problem. Annals of Operations Research 179(1), 77–103.

Hassan, M. M., & Hogg, G. L. (1991). On constructing a block layout by graph theory. The international journal of production research, 29(6), 1263-1278.

Hassan, M. M., Hogg, G. L., & Smith, D. R. (1986). SHAPE: a construction algorithm for area placement evaluation. International Journal of Production Research, 24(5), 1283-1295.

Hathhorn, J., Sisikoglu, E., & Sir, M. Y. (2013). A multi-objective mixed-integer programming model for a multi-floor facility layout. International Journal of Production Research, 51(14), 4223-4239.

HC de la Santé Publique. Rapport sur la sécurité anesthésique. 1994. Paris: Éditions Ecole Nationale de la Santé Publique.

Helber, S., Böhme, D., Oucherif, F., Lagershausen, S., & Kasper, S. (2014). A hierarchical facility layout planning approach for large and complex hospitals (No. 527). Discussion Paper, Wirtschaftswissenschaftliche Fakultät, Leibniz Universität Hannover.

Holland J (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor.

Huang, C., & Wong, C. K. (2015). Optimization of site layout planning for multiple construction stages with safety considerations and requirements. Automation in Construction, 53, 58-68.

I.H. Osman (2006). A tabu search procedure based on a ranked-roulette diversification for the weighted maximal planar graph problem. Computers and Operations Research.

Jaafari, A. A., Krishnan, K. K., Doulabi, S. H. H., & Davoudpour, H. (2009). A multi-objective formulation for facility layout problem. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 2).

Jennings, N.R., Sycara, K. & Wooldridge, M. (1998). A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems, 1, 7–38. Cited on pages 1, 2, and 3.

Jiang, S., & Nee, A. Y. C. (2013). A novel facility layout planning and optimization methodology. CIRP Annals-Manufacturing Technology, 62(1), 483-486.

Jolai, F., Tavakkoli-Moghaddam, R., & Taghipour, M. (2012). A multi-objective particle swarm optimisation algorithm for unequal sized dynamic facility layout problem with pickup/drop-off locations. International Journal of Production Research, 50(15), 4279-4293.

Kia, R., Paydar, M. M., Jondabeh, M. A., Javadian, N., & Nejatbakhsh, Y. (2011). A fuzzy linear programming approach to layout design of dynamic cellular manufacturing systems with route selection and cell reconfiguration. International Journal of Management Science and Engineering Management, 6(3), 219-230.

Kim, D. G., & Kim, Y. D. (2010). A branch and bound algorithm for determining locations of long-term care facilities. European Journal of Operational Research, 206(1), 168-177.

Kim, D. G., & Kim, Y. D. (2013). A Lagrangian heuristic algorithm for a public healthcare facility location problem. Annals of Operations Research, 206(1), 221-240.

Kim, D. G., Kim, Y. D., & Lee, T. (2012). Heuristics for locating two types of public health-care facilities. Industrial Engineeering & Management Systems, 11(2), 202-214.

Kim, J. Y., & Kim, Y. D. (1995). Graph theoretic heuristics for unequal-sized facility layout problems. *Omega*, *23*(4), 391-401.

Konak, A., Kulturel-Konak, S., Norman, B. A., & Smith, A. E. (2006). A new mixed integer programming formulation for facility layout design using flexible bays. Operations Research Letters, 34(6), 660-672.

Koopmans, T.C., and M. Beckman (1957). Assignment problems and the location of economic activities. Econometric 25, 53-76.

Kothari, R., & Ghosh, D. (2013). Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods. European Journal of Operational Research, 224(1), 93-100.

Krishna, K. K., Amir, A. J., Abolhasanpour, M., & Hosein H. (2009). A mixed integer programming formulation for multi-floor layout. African Journal of Business Management Vol. 3 (10), pp. 616-620.

Kulturel-Konak, S. (2007). Approaches to uncertainties in facility layout problems: Perspectives at the beginning of the 21st Century. Journal of Intelligent Manufacturing, 18(2), 273-284.

Kulturel-Konak, S., & Konak, A. (2011). Unequal area flexible bay facility layout using ant colony optimisation. International Journal of Production Research, 49(7), 1877-1902.

Kusiak, A. (1990). Intelligent manufacturing systems. New Jersey: Prentice-Hall International Inc.

Kusiak, A., & Heragu, S. S. (1987). The facility layout problem. European Journal of operational research, 29(3), 229-251.

Lacksonen, T.A., & Enscore, E.E. (1993). Quadratic assignment algorithms for the dynamic layout problem. International Journal of Production Research.313, 503–17.

Lahmar, M., & Benjaafar, S. (2005). Design of distributed layouts. IIE Transactions, 37(4), 303-318.

Lawler, E. L. (1963). The quadratic assignment problem. Management science,9(4), 586-599.

Leung, J. (1992). A new graph-theoretic heuristic for facility layout. Management Science, 38(4), 594-605.

Lin, Q. L., Liu, H. C., Wang, D. J., & Liu, L (2013). Integrating systematic layout planning with fuzzy constraint theory to design and optimize the facility layout for operating theatre in hospitals. Journal of Intelligent Manufacturing, 1-9.

Liu, Q., & Meller, R. D. (2007). A sequence-pair representation and MIP-model-based heuristic for the facility layout problem with rectangular departments. IIE Transactions, 39(4), 377-394.

Lorenz, W. E., Bicher, M., & Wurzer, G. (2015, February). Adjacency in Hospital Planning. In Mathematical Modelling (Vol. 8, No. 1, pp. 862-867).

Madhusudanan Pillai, V., Hunagund, I. B., & Krishnan, K. K. (2011). Design of robust layout for dynamic plant layout problems. Computers & Industrial Engineering, 61(3), 813-823.

Maniezzo V., & Colorni A. (1999). The ant system applied to the quadratic assignment problem. IIE Trans Knowledge Data Eng 11(5):769–778

Marzetta, A., & Brüngger, A. (1999). A dynamic-programming bound for the quadratic assignment problem. In Computing and Combinatorics (pp. 339-348). Springer Berlin Heidelberg.

Mazinani, M., Abedzadeh, M., & Mohebali, N. (2013). Dynamic facility layout problem based on flexible bay structure and solving by genetic algorithm. The International Journal of Advanced Manufacturing Technology, 65(5-8), 929-943.

McKendall Jr, A. R., & Liu, W. H. (2012). New Tabu search heuristics for the dynamic facility layout problem. International journal of production research, 50(3), 867-878.

Meller, R. D., & Gau, K. Y. (1996a). Facility layout objective functions and robust layouts. International journal of production research, 34(10), 2727-2742.

Meller, R. D., & Gau, K. Y. (1996b). The facility layout problem: recent and emerging trends and perspectives. Journal of manufacturing systems, 15(5), 351-366.

Miranda, G., Luna, H.P.L., Mateus, G.R., Ferreira, R.P.M., 2005. A performance guarantee heuristic for electronic components placement problems including thermal effects. Computers and Operations Research 32, 2937–2957.

Moslemipour, G., Lee, T. S., & Rilling, D. (2012). A review of intelligent approaches for designing dynamic and robust layouts in flexible manufacturing systems. The International Journal of Advanced Manufacturing Technology, 60(1-4), 11-27.

Motaghi, M., Hamzenejad, A., & Riyahi, L. (2011). Optimization of hospital layout through the application of heuristic techniques (Diamond Algorithm) in Shafa hospital. International Journal of Management of Business Research, 1(3), 133–138.

Nyberg, A., & Westerlund, T. (2012). A new exact discrete linear reformulation of the quadratic assignment problem. European Journal of Operational Research, 220(2), 314-319.

Nyberg, A., Westerlund, T., & Lundell, A. (2013). Improved discrete reformulations for the quadratic assignment problem. In Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (pp. 193-203). Springer Berlin Heidelberg.

Osman, I. H., Hasan, M., & Abdullah, A. (2002). Linear programming based meta-heuristics for the weighted maximal planar graph. Journal of the Operational Research Society, 1142-1149.

Osman, I.H., Al-Ayoubi, B., and Barake, M.A. (2003). A greedy random adaptive search procedure for the weighted maximal planar graph problem. Computers and Industrial Engineering. Vol. 45, 635-651.

Papageorgiou, L. G., & Rotstein, G. E. (1998). Continuous-domain mathematical models for optimal process plant layout. Industrial & engineering chemistry research, 37(9), 3631-3639.

Pardalos, P.M., F. Rendl, and H. Wolkowicz (1994). The Quadratic Assignment Problem: A Survey and Recent Developments, in Quadratic Assignment and Related Problems, P.M.

Patsiatzis, D. I., & Papageorgiou, L. G. (2002). Optimal multi-floor process plant layout. Computers & Chemical Engineering, 26(4), 575-583.

Pourvaziri, H., & Naderi, B. (2014). A hybrid multi-population genetic algorithm for the dynamic facility layout problem. Applied Soft Computing, 24, 457-469.

Rahman, S., Smith, D.K., 2000. Use of location-allocation models in health service development planning in developing nations. European Journal of Operational Research 123, 437–452.

Rosenblatt, M. J. (1986). The dynamics of plant layout. Management Science, 32(1), 76–86.

Şahin, R. (2011). A simulated annealing algorithm for solving the bi-objective facility layout problem. Expert Systems with Applications, 38(4), 4460-4465.

Sahni, S. and T. Gonzalez (1976). P-complete Approximation Problems. Journal of the ACM 23, 1976, 555-565.

Saidi-Mehrabad, M., & Safaei, N. (2007). A new model of dynamic cell formation by a neural approach. The International Journal of Advanced Manufacturing Technology, 33(9-10), 1001-1009.

Samarghandi, H., Taabayan, P., & Jahantigh, F. F. (2010). A particle swarm optimization for the single row facility layout problem. Computers & Industrial Engineering, 58(4), 529-534.

Seppänen, J., & Moore, J. M. (1970). Facilities planning with graph theory. Management Science, 17(4), B-242.

Sha, D. Y., & Chen, C. W. (2001). A new approach to the multiple objective facility layout problem. Integrated Manufacturing Systems, 12(1), 59-66.

Shafigh, F., Defersha, F. M., & Moussa, S. E. (2015). A mathematical model for the design of distributed layout by considering production planning and system reconfiguration over multiple time periods. Journal of Industrial Engineering International, 1-13.

Singh, S. P., & Sharma, R. R. (2006). A review of different approaches to the facility layout problems. The International Journal of Advanced Manufacturing Technology, 30(5-6), 425-433.

Singh, S. P., & Singh, V. K. (2010). An improved heuristic approach for multi-objective facility layout problem. International Journal of Production Research, 48(4), 1171-1194.

Solimanpur, M., & Jafari, A. (2008). Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. Computers & Industrial Engineering, 55(3), 606-619.

Syam, S. S., & Côté, M. J. (2012). A comprehensive location-allocation method for specialized healthcare services. Operations Research for Health Care, 1(4), 73-83.

Talukder, S. (2011). Mathematical modelling and applications of particle swarm optimization (Doctoral dissertation, Blekinge Institute of Technology).

Tarkesh, H., Atighehchian, A., & Nookabadi, A. S. (2009). Facility layout design using virtual multi-agent system. Journal of Intelligent Manufacturing, 20(4), 347-357.

Tian, Z., Li, H., & Zhao, Y. (2010). Optimization of continuous dynamic facility layout problem with budget constraints. In Information Science and Engineering (ICISE), December, 2010 2nd International Conference on (pp. 387-390). IEEE.

Ulutas, B. H., & Islier, A. A. (2009). A clonal selection algorithm for dynamic facility layout problems. Journal of Manufacturing Systems, 28(4), 123-131.

Ulutas, B. H., & Kulturel-Konak, S. (2012). An artificial immune system based algorithm to solve unequal area facility layout problem. Expert Systems with Applications, 39(5), 5384-5395.

Urban, T. L. (1993). A heuristic for the dynamic facility layout problem. IIE transactions, 25(4), 57-63.

Vos, L., S. Groothuis, and G. G. van Merode (2007). Evaluating hospital design from an operations management perspective. Health Care Management Science 10(4), 357–364.

Wahed, M. A., Ouda, B. K., & Elballouny, M. (2011). Evaluation of an Operating Theater Design by a Software Program (OTDA: Operating Theater Design Analyzer). International Journal of Database Management Systems, 3(3).

Wooldridge, M., Jennings, N. R. Intelligent agents: Theory and practice. The Knowledge Engineering review, vol. 10(2), pp: 115-152, 1995.

Yeh, I. C. (2006). Architectural layout optimization using annealed neural network. Automation in construction, 15(4), 531-539.

Zhang, Y., & Che, A. (2014, November). A Mixed Integer Linear Programming approach for a new form of facility layout problem. In Control, Decision and Information Technologies (CoDIT), 2014 International Conference on (pp. 065-068). IEEE.

Shouman, M.A., et al. Facility Layout Problem (FLP) and Intelligent Techniques: A Survey. In Proceedings of 7th International Conference on Production Engineering, Design and Control. 2001. Alexandria, Egypt.