



An Interactive-based approach to the Layout design optimization

Julien Bénabès, Fouad Bennis, Emilie Poirson, Yannick Ravaut

► To cite this version:

Julien Bénabès, Fouad Bennis, Emilie Poirson, Yannick Ravaut. An Interactive-based approach to the Layout design optimization. 20th CIRP Design Conference, Apr 2010, NANTES, France. hal-00512470

HAL Id: hal-00512470

<https://hal.science/hal-00512470>

Submitted on 30 Aug 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Interactive-based approach to the Layout design optimization

Julien Bénabès⁽¹⁾, Fouad Bennis⁽¹⁾, Emilie Poirson⁽¹⁾ and Yannick Ravaut⁽²⁾

⁽¹⁾ Ecole Centrale de Nantes, Irccyn, UMR CNRS 6597, Nantes, France,
julien.benabes@irccyn.ec-nantes.fr

⁽²⁾ Thales Communications, Cholet, France,
Yannick.RAVAUT@fr.thalesgroup.com

Abstract

Layout design plays an important role in the design and usability of many engineering products and systems. Because of the great complexity of most of industrial layout problems, the decision of the acceptable layout is a hard and critical task since the special layout can have a significant consequence on the global performances. Thus, in order to propose to the designer an optimal spatial arrangement in a reasonable time, this paper develops an interactive optimization strategy that is tested on the facilities layout problem of a shelter. This problem is innovative because it introduces the concept of space of accessibility.

Keywords:

Layout problem, Interactive optimization, Genetic algorithm.

1 INTRODUCTION

Layout problem is inherently a multidisciplinary task [1]. It covers all the aspects of the product design life cycle from the conceptual to the detailed stage and makes necessary the collaboration between experts of technical and economical disciplines. In fact, layout design is usually formulated as an optimization problem: *find the best arrangement (location and orientation) of components in a given available space satisfying geometrical and functional constraints*. A non-overlapping constraint is basically a common geometrical constraint for all 3-Dimensional Layout problems, while alignment, orientation or gathering components refer to functional constraints. Because of the geometrical complexity, the 3-Dimensional layout problem optimisation is non-linear and NP-hard. It means that the problem is intrinsically harder than those that can be solved by a nondeterministic Turing machine in polynomial time. The objective and constraints evaluation is generally time consuming.

It is essential to distinguish between Cutting and Packing (C&P) problems and 3-Dimensional Layout problems. In C&P problems, components are only geometrically related to each other, whereas in layout problems, components are geometrically and functionally related and connected. This difference leads different tools and methods to solve each class of problem being aware of the common non-overlap constraints in the two problems.

Typologies of C&P problems have been proposed [2] but as far as we know, there is no general typology of layout problems. Amine Drira has described a tree representation of facility layout problems [3] that depends on design constraints and objectives of the location of facilities inside a plant.

Actually, layout problems can be divided into several kinds of specific problems, which have their own solving method. C&P problems can be assimilated as a particular application of each specific problem.

Layout problems can also be classified according to three criteria: the compactness of the problem, the number and type of design constraints and objectives and the geometrical complexity of the design components. For example, the container loading problem and the engine

layout design are not in the same category of problem. It means that, for the two problems, the compactness is important but in the engine layout design, constraints and objectives are multiple (non-overlap and functional constraints, accessibility objective...) and the different parts of the engine have complex lines. The container loading problem does not have the same characteristics.

The formulation of layout problems uses mono-objective or multi-objective optimisation. The designer can make an early decision by using an aggregation function in order to transform a multi-objective into a mono-objective one. This approach is only effective when all data and information on the aggregation are available or if the designer is familiar with the specific layout problem. In this paper, we use multi-objective optimization. The decision on the preferences between objective functions is delayed so that the designer can use the Pareto-front in order to select the most appropriate solution. In this approach, the designer has to simultaneously optimize two or more conflicting objectives subject to constraints.

The general formulation of an optimization problem can be written as the equation 1, where m is the number of objective functions and n the number of design variables.

$$\begin{cases} \text{find the design variables } \mathbf{x} = (x_1, x_2, \dots, x_n) \\ \min F(\mathbf{x}) = (f_1, f_2, \dots, f_m) \\ \text{s.t. } g(\mathbf{x}) \leq 0 \text{ and } h(\mathbf{x}) = 0 \end{cases} \quad (1)$$

The designer has to compare two solutions represented by two vectors of objectives $F(U) = (f_{1U}, f_{2U}, \dots, f_{mU})$ and $F(V) = (f_{1V}, f_{2V}, \dots, f_{mV})$ where f_{iU} is the i th component of the vector of objectives F for the design variable U . In fact, U dominates V (Pareto dominance) if U is as good as V for all the objectives and U is better than V for at least one objective. Mathematically, this can be formulated by:

$$\begin{cases} \forall i \in [1, \dots, m] \quad f_{iU} \leq f_{iV} \\ \exists j \in [1, \dots, m] \quad f_{jU} < f_{jV} \end{cases} \quad (2)$$

Multi-objective optimization searches for the set of non-dominated points (assimilated to Pareto-optimal points in

the next sections of this paper) in the objective space given by efficient solutions. Figure 1 represents the Pareto front for an optimization problem defined by two objectives ($\min f_1, \min f_2$) where U dominates V.

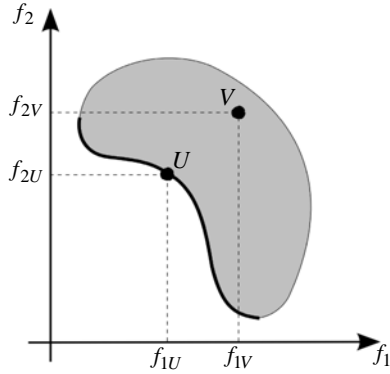


Figure 1: Pareto front of a multi-objective problem.

One finds multiple search algorithms to solve layout optimization problems in two or three dimensions. Traditional optimization approaches for three dimensional layout problems are described by Jonathan Cagan [4]. They use genetic algorithms [5], simulated-annealing algorithms [6,7] or extended pattern search algorithms [8]. Most search algorithms are developed for a specific problem and they provide an effective optimization strategy for this kind of problem. Therefore, they are not generic and can not be adapted to other layout problems. In this paper, the proposed method is based on a generic technique for solving layout problems. The design strategy uses a Genetic Algorithm coupled with a Separation Algorithm, in order to ensure a good diversity of solutions computed by the algorithm, and allows the designer to interact with the Pareto-optimal solutions.

This paper is organized as follows: in section 2, the synopsis of the proposed optimization method is presented. In section 3, the proposed method is tested on the layout problem of facilities inside a shelter. Optimization problem formulation and results obtained by the method are described and analyzed. Sections 4 and 5 are dedicated to an outlook on future work and the conclusion.

2 PROBLEM SOLVING STRATEGY

The constraint space of design variables is highly composite because of the geometric complexity, the non-overlap constraints and the relative location between the components. This property leads us to recognise that there is no choice but using stochastic or heuristic techniques for a 2-D or 3-D layout problem. This kind of technique makes it possible to explore efficiently the design space and avoid a local optimum. We also recognise that a multi-objective Genetic algorithm is suitable for this problem.

Basically, a genetic algorithm search uses the mechanics of natural selection and natural genetics to evolve a population of initial solutions into a near-optima solution. The common idea behind these algorithms is the same: given a set of individuals from an initial population, the genetic algorithm uses basically 3 operators in order to create a set of new candidate solutions. This process can iterate until candidates with sufficient good fitness are obtained. The three operators are namely: selection, mutation and crossover or recombination. Mutation and recombination act to create the necessary diversity and novelty and selection acts as a force of pushing quality (fitness). Repeated selection from the same population would produce nothing better than multiple copies of the

best individual originally in it. For improvement to be able to occur, some novelty must be introduced into the population between selection steps. The genetic operators modify the selected parents by manipulating their genotype.

Since the genetic algorithm is based on stochastic operators and parameters, the progression of multi-objective layout optimisation is time consuming. It depends also on the number of design variables and the number of components and the types of design constraints. In order to improve the algorithm we introduce two new steps into the global process of the genetic algorithm: separation techniques and interaction with the designer. The initial random population of the GA is improved using the separation technique and interaction. In fact, the initial random population of the GA leads to a high number of overlap components thus the GA fails to find efficient candidates.

The objective of the search algorithm proposed in this paper is to generate a uniformly distributed global Pareto front for layout optimization problems. Our strategy consists of initializing the multi-objective optimizer with a population of individuals which have been locally modified by a separation algorithm and designer interaction in order to reduce the violation of placement constraints. Thus, the strategy is based on three complementary approaches, which are clearly separated:

1. Firstly, the generation of a database of mixed designs that respect non-overlap constraints.
2. Secondly, the optimization of this database by considering all the design objectives (with a Genetic Algorithm).
3. At the end of the GA iteration process, the interactive choice of the suitable solution by the designer is conducted using the non dominated Pareto front generated by the GA.

2.1 Separation Algorithm

Several separation algorithms have been proposed [9,10]. However, the key idea is always the same: given a configuration that doesn't satisfy location constraints, the objective of the separation algorithm is to minimize the non-respect of overlap between components and protrusion (overlap between components and the non allowed space).

For solving simple layout problems in two dimensions, the separation problem is formulated as an unconstrained minimization problem defined by:

$$(\text{Sep Algo}) \left\{ \begin{array}{l} \min F(x) = \sum A_{ij} \\ i, j \in [1, \dots, n], i \neq j \end{array} \right. \quad (3)$$

where A_{ij} represents the intersection area between the components i and j . Consequently, it is possible to define the violation of placement constraints F as the total sum of intersection areas between the different elements which make up the layout design. For example, let us consider that all the items of the layout design are rectangles, the intersection area between two items is equal to:

$$A_{ij} = \max[0, \min(x_i + \frac{L_i}{2}, x_j + \frac{L_j}{2}) - \max(x_i - \frac{L_i}{2}, x_j - \frac{L_j}{2})] \\ \times \max[0, \min(y_i + \frac{L_i}{2}, y_j + \frac{L_j}{2}) - \max(y_i - \frac{L_i}{2}, y_j - \frac{L_j}{2})] \quad (4)$$

where (x_i, y_i) are the coordinates of the geometric center of the rectangle i . L_i and l_i represent respectively the length and the width of the rectangle i .

The algorithm used to minimize F is based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. This algorithm computes a finite-difference approximation of the gradient and the hessian of the function F in order to locally modify the optimization variables and to minimize F . The algorithm stops after a certain number of iterations.

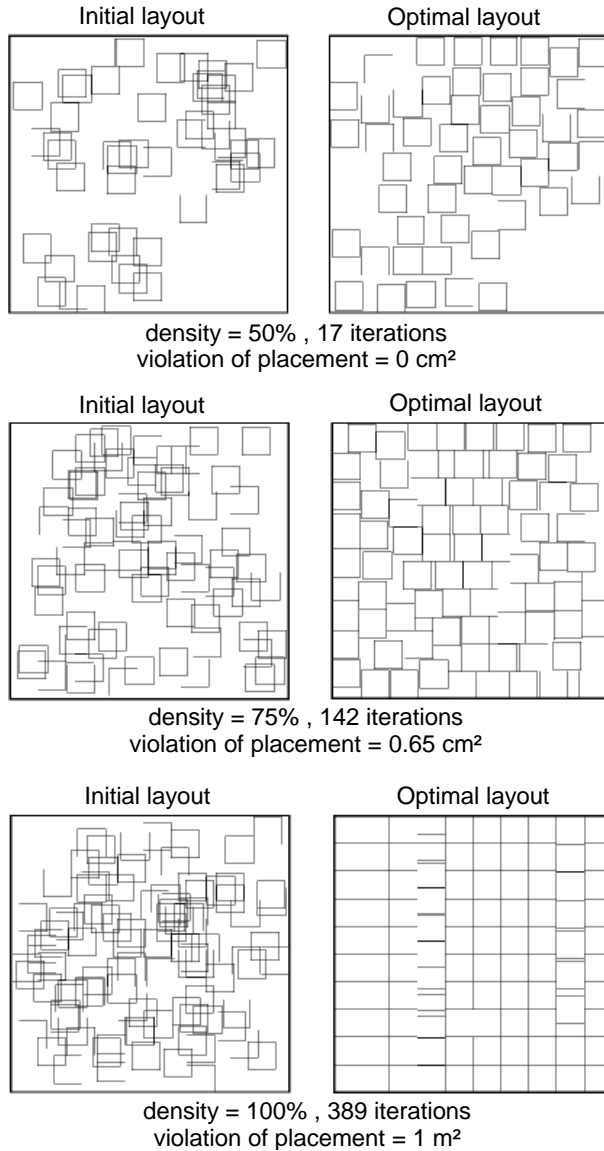


Figure 2: Separation algorithm BFGS test.

In order to understand the principle of the separation algorithm, let us consider a layout problem test. The dimensions of the square container are $10 \times 10 = 100 \text{ m}^2$. The objective of this 2D-configuration problem is to place N square items whose dimensions are $1 \times 1 = 1 \text{ m}^2$ in the container. It means that the algorithm searches the optimal configuration that reduces the violation of placement constraints F , which has been previously defined as the total sum of intersection areas between the square items and the container. Figure 2 shows simulations results, considering different values of the problem density.

2.2 Multi-objective optimization

The multi-objective optimizer is in charge of exploring efficiently the search space to propose trade-off solutions. The proposed method uses the Multi-Objective Genetic Algorithm (MOGA-II) [11]. MOGA-II is an efficient Multi-Objective Genetic Algorithm (MOGA) [12] that uses a

smart multi-search elitism. This new elitism operator is able to preserve some excellent solutions without bringing premature convergence to local-optimal frontiers. For simplicity, MOGA-II requires only very few user-provided parameters. Several other parameters are internally settled in order to provide robustness and efficiency to the optimizer. Three genetic operators are used to generate new solutions:

- **Directional Cross-over:** the cross-over operator is a method of recombination where parents produce offspring by sharing information. The aim of this operator is to obtain individuals with better characteristics while maintaining the diversity of the population.
- **Selection:** this value gives the probability that design configurations are not changed during the evolution. In order to maintain a good diversity between points, this parameter should be kept small,
- **Mutation:** this value gives the probability that a design configuration is randomly changed.

For the application studied in section 3, the genetic operators have been set as shown in Table 1. The number of individuals in the initial generation is equal to 240 because a rule of thumb suggests the possibly to accumulate an initial population of at least 16 design configurations and possibly more than $2 \times \text{Number of variables} \times \text{Number of objectives} = 2 \times 24 \times 5 = 240$.

Number of individuals in the initial generation	240
Probability of Directional Cross-Over	0.1
Probability of Selection	0.05
Probability of Mutation	0.45
Probability of Classical Cross-Over	0.4
Elitism	enabled
DNA String Mutation Ratio	0.5
Number max of generations	100

Table 1: Multi-Objective Genetic Algorithm parameters.

2.3 Interactive process

In general the development of an engineering object is considered as a single process involving multicriteria identification of the mathematical model followed by multicriteria optimization of the object design on the basis of this mathematical model. The process of statement-solution of engineering design problems without the interference of the design is impossible. In solving the design problem, the designer almost always has to correct either the mathematical model, the dimension of the vectors of design variables and criteria, the design variable ranges, and so on. This creative process of correcting an initial statement is natural when solving engineering problems. The direct participation of the designer in the construction of the feasible design and nonformal analysis are the essential stage of the search for the optimal design [13]. The simulation tools provide powerful solutions for planning and designing of complex mechanical systems. The problem with these is the representation and the interpretation of the results by the engineer. Important for the engineer is not only the value of the point but its variation and the information about the most favorable directions. The exploitation of the results is not obvious and the link with the performance value of the real phenomenon is not trivial. When one analyzes the communication between the operator and the computer, it

can be perceived that the operator immersion in the digital model is very weak.

It was pointed out in section 1, that the decision can be made earlier in the design optimisation process by the creation of an aggregation of the objective function or later using the non-dominated points generated by the genetic algorithm. The proposed strategy uses two interactive steps. The first one is the interactivity of the designer for the selection of feasible solution to present as an initial population of the genetic algorithm. The first interaction step is limited to the geometrical non-overlap of components. The GA algorithm can be stopped after a fixed number of iterations. Since all the non-dominated points are potentially good acceptable solutions of the given Layout problem, the designer has to explore the set of these points and select the best solution. It is well recognised by the expert of the optimisation that it is always very hard to express all the designer requirements using only the objective function. Several subjective functions are qualitative and can not easily be expressed explicitly by using a numerical value. In order to take into account this subjective aspect of the layout problem, selected solutions of the non-dominated front are presented to the designer and an interactive numerical environment is used to support the decision. Since the number of non-pareto points can be to very large, we also use a reduction approach that regroups all the equivalent solutions. Then for each presented solution, the designer can act on the environment and locally modify the position or orientation of the appropriate components. This step is different to the initial interaction since the designer has also the evaluation of all the objective function. The interactivity is not only limited to the geometrical visual evaluation but the value of the objectives function and there deviation are helpful at this step.

3 APPLICATION

3.1 Problem description

In this section we consider an application of the proposed strategy to find the optimal layout of facilities in a shelter. Several components have to be arranged in the shelter including electric and energetic cabinets, desks and electrical boxes. The CAD model of the shelter is presented Figure 3.

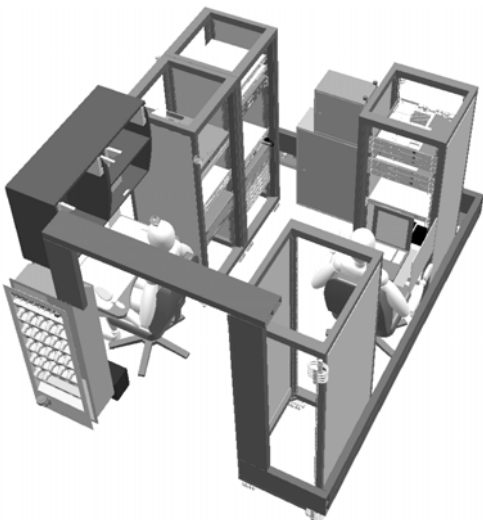


Figure 3: Overall view of the shelter.

The layout optimization of this shelter is a three dimensional optimization problem. However for the presentation, and fortunately because the cabinets are the full height of the shelter and prevent a superposition of elements, the model is simplified and conceptualized in

two dimensions. The simplified model of the shelter is shown Figure 4.

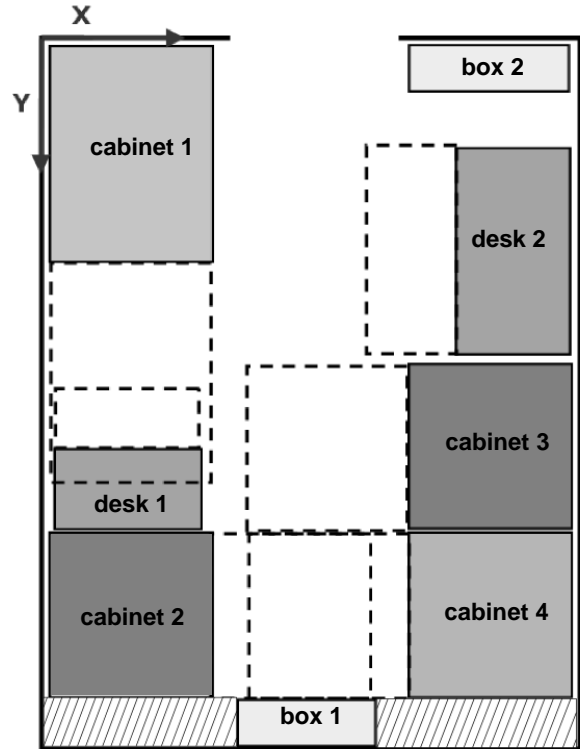


Figure 4: Configuration model of the shelter in 2D.

The formulation of this layout problem is innovative because the components can be classified in two categories: those which have a mass (*material components*) and those which no have mass (*virtual components*). Here, the virtual components represent the spaces of accessibility of the cabinets and the desks. For example, the space of accessibility of the cabinet is the required space to insert some materials into the cabinet. These spaces are symbolized in Figure 4 by dotted rectangles. With this problem formulation, design constraints depend on the category of components. It means that overlap is allowed between two spaces of accessibility, taking account that operations of materials loading are sequentially made whereas overlap has to be minimized between two material components.

Moreover, the space, represented by hatching in Figure 4, is the space below the air-conditioner where no cabinet can be placed. This space is also a virtual component that is fixed during the optimization process

The dimensions of the shelter are 2150mm x 2740mm. The density of this configuration, without considering the spaces of accessibility of the different components, is equal to 50%. If the spaces of accessibility are considered, this density increases up to 90%.

3.2 Problem formulation

Problem formulation is a very important step of the optimization process. The optimization problem studied here is an under constrained multi-objective problem. Let us see how the variables, constraints and design objectives are defined.

Optimization variables

Each layout component has three optimization variables (X, Y, α): the coordinates of each element (a continued variable along X axis and another one along Y axis) and the rotation angle (one discrete variable along Z axis).

Consequently, the number of optimization variables for this problem is equal to 24 (= 8 items x 3 coordinates). Because of the rotation of each component, variables X and Y are bounded according to the following relation (for the variable X_i for example):

$$\min(l_i, L_i) < X_i < l_{sh} - \min(l_i, L_i) \quad (5)$$

where l_{sh} represents the width of the shelter. Here, l_i is the dimension of the component along X axis (it does not have to be confused as the width of the component i).

Design constraints

The design constraints of this layout problem are non-overlap constraints. They are divided in four categories, according to the following classification:

- Non-overlap constraints between components (C1).
- Non-overlap constraints between components and spaces of accessibility (dotted rectangle represented in Figure 4) (C2).
- Non-protrusion constraints between components, spaces of accessibility and the shelter (C3).
- Non-overlap constraints between cabinets and the space below the air-conditioner (hatching represented in Figure 4) (C4).

The rectangular shape of components simplifies the formulation of design constraints. Thus, the non-overlap constraint between the rectangles i and j is equal to the intersection area between the component i and j (in cm²). This area has been defined in equation 4. Actually, the objective of the separation algorithm is defined as:

$$F = C1 + C2 + C3 + C4 \quad (6)$$

Design objectives

In collaboration with the company experts of this specific problem, we have considered for this optimization problem the five following design objectives:

- To minimize the distance between the center of gravity of components and the geometrical center of the shelter, in order to balance the masses inside the shelter (O1).
- To maximize the distance between the cabinet 1 and the cabinets 2 and 3 and the electrical box 2, in order to limit interactions between energy and electric network (O2, O3, O4).
- To minimize the distance between the electrical box 2 and one of the shelter's walls, in order to establish a connection with exterior (O5).

The design objectives O2, O3, O4 and O5 are formulated by the distance between the centers of gravity of elements. For example, the distance d_{ij} between the components i and j is equal to:

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (7)$$

where x and y are the coordinates of the centers of gravity of the items i and j.

Let us consider the coordinates of the center of gravity of all the elements which are placed in the shelter. These coordinates are equal to:

$$X_{gra} = \frac{\sum_{i=1}^N (x_i \times m_i)}{\sum_{i=1}^N m_i}, \quad Y_{gra} = \frac{\sum_{i=1}^N (y_i \times m_i)}{\sum_{i=1}^N m_i} \quad (8)$$

where N is equal to the number of elements which have a mass: the cabinets, the desks, the electrical boxes and the air-conditioners. Then, by considering equation 7, the objective 1 (O1) is computed.

More designer's knowledge could be incorporated in the layout problem formulation. It means for example, in the configuration design of the shelter studied in this paper, the design objective O5 can be deleted and the degree of freedom of the electrical box 2 can be reduced, in order to force it to displace along one of the walls of the shelter. The designer's contribution for formulation should simplify the search of feasible solutions by reducing the number of possible solutions.

3.3 Results and analysis

The resolution of this optimization problem has been firstly realized only with the multi-objective optimizer MOGA-II. The algorithm has been randomly initialized with a population of 240 designs. Most of these initial designs did not respect the non-overlap constraints because they have been randomly generated. Because of the great density of this layout problem, only one or two feasible configurations were computed for each simulation. A configuration is defined as:

Design j is a new configuration if it differs from the design i by at least one of the following criteria:

- One of the components of the layout has been displaced from at least Δ mm along one of the axis X or Y, (Δ is set to 500 mm in this application).
- One of the components has been rotated.
- The minimum difference between the objective values of the two designs is bigger than a limit, for example, fixed at 10 cm.

These results lead us to use the method proposed in this paper in order to generate, with only one optimization simulation, a set of well distributed Pareto-optimal designs.

Thus, the results obtained for each step of the method are described here:

1. **Separation algorithm and first interaction with the designer:** the algorithm has been randomly initialized with designs that do not respect non-overlap constraints. Then, a set of 162 "feasible" (it means that respect design constraints) designs have been computed. This population then has been completed with 78 individuals randomly generated in order to create the first population (240 individuals) and to guarantee the diversity of the genetic algorithm.
2. **Multi-Objective Genetic Algorithm:** the algorithm has searched optimal solutions by considering the design constraints and all the objectives of the problem. Then, after a hundred generations, a set of 172 feasible configurations have been computed. 41 of these solutions are Pareto-optimal designs. Figure 5 shows three of these Pareto-optimal configurations and the initial solution. It's important to mention that this initial configuration is an intuitive solution which has been generated only by considering geometric aspects.

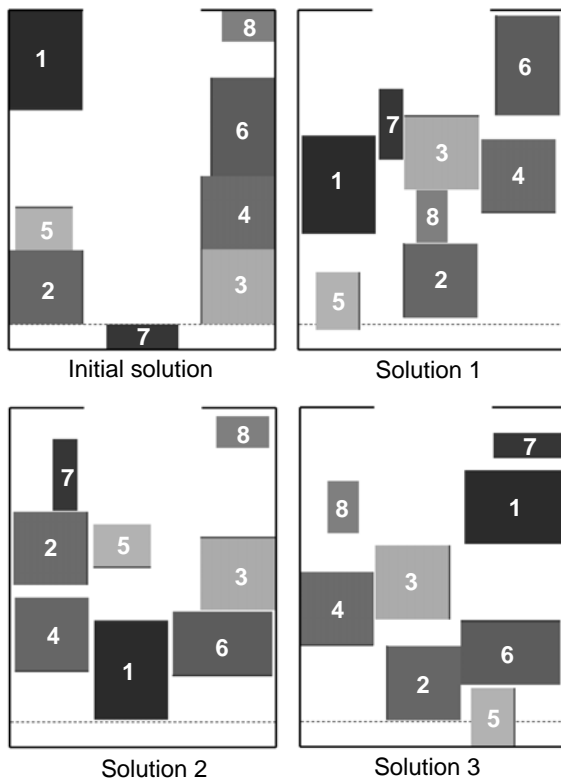


Figure 5: Solution computed by the algorithm.

The Pareto-optimal solutions, shown in Figure 5, point out a problem that is present in all the solutions computed by the algorithm. In fact, Figure 5 shows that, even though the design is a feasible configuration that respects all the design constraints, some facilities (for example facilities n°2, 4, 5, 8 in the first solution) are not accessible from the shelter's entry. It means that a design constraint or an objective is missing in the layout problem formulation, in order to characterize the accessibility to a facility from the shelter's entry.

Consequently, the problem description has been changed and a free corridor, located in the middle of the shelter, has been added. In fact, this corridor is a space of living that can be considered as a fixed virtual component, where all the material components can not be placed. This space is going to resolve the problem of accessibility inside the shelter.

Then, with the separation algorithm, a set of feasible designs have been computed and by interacting directly with them and by relaxing the design constraints (until 150 cm²), the designer has selected 78 different designs. This population then has been completed with 162 individuals randomly generated. Next, the genetic algorithm has generated 14 feasible configurations whose 7 are Pareto-optimal solutions.

3. **Interactive decision making:** the 7 Pareto-optimal designs do not dominate the initial solution. On the other hand, the initial solution does not dominate either the solutions computed by the proposed method. Actually, it means that the designer is the only person who can make the final design choices.

In order to make a decision, the method provides to the designer an interactive geometric and numeric visualization of the designs. The designer can explore the set of non-dominated solutions, compare their objective values and interact with them. Figure 6 represents the interface that allows the designer to visualize at the same time the layout design and the associated design constraints and objectives values. The designer can also compare two solutions. In fact,

when the designer displaces one component or changes its direction, the design constraints and objectives are automatically actualized.

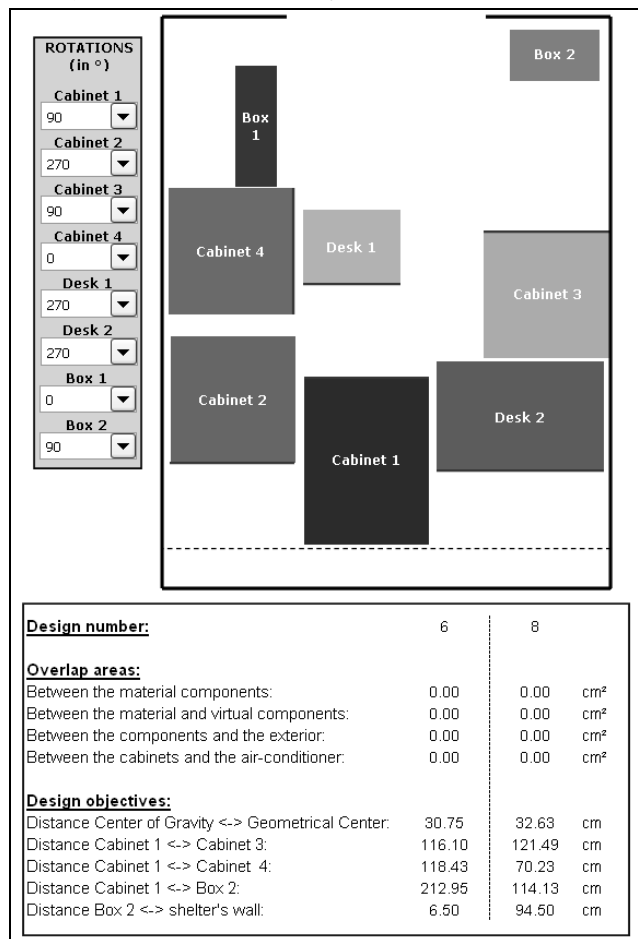


Figure 6: Interactive layout interface.

Actually, among the 7 Pareto-optimal solutions computed by the algorithm, let us focus on the third solution. Locally changing the location of some components of this design improves its performances. Figure 7 shows this solution 3 and the solution that results from the modifications made by the designer.

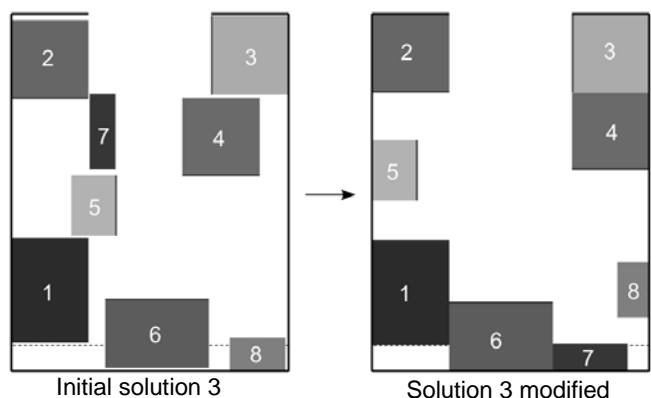


Figure 7: Local modification of the solution 3.

Table 3 describes all the objectives values for the initial solution generated by the expert, the solution 3 and the solution 3 locally modified by the designer. Actually, due to the interaction of the designer with the final solutions, a new solution, better than the initial one, is created.

Design objective	Initial solution	Solution 3	Improved solution 3
O1 (minimize)	25.41 cm	8.67 cm	3.48 cm
O2 (maximize)	240.58 cm	238.30 cm	240.58 cm
O3 (maximize)	198.50 cm	177.80 cm	198.50 cm
O4 (maximize)	165.80 cm	168.29 cm	172.51 cm
O5 (minimize)	0 cm	0 cm	0 cm

Table 3: Industrial solution vs solution 3.

4 RESTROSPECT AND PERSPECTIVE

This paper has introduced a new interactive optimization strategy for solving layout problems. The method can be divided into several steps, as shown in Figure 8. Firstly, a population of designs, randomly initialized, is optimized by the separation algorithm. Then, the designer interacts with the solutions computed by the algorithm and selects some individuals according to the design constraints. Secondly, the new population is optimized by the multi-objective optimizer by considering all the design objectives. Then, the designer can locally modify some computed designs in order to improve their objectives. Actually, our strategy has the innovative particularity to allow the designer to interact with the optimization process in order to improve the performances of the Pareto-optimal designs and to keep a good diversity in computed solutions.

The application, which has been studied in this paper, has an innovative problem formulation because it introduces the concept of space of accessibility, which can be considered as a virtual component of the layout design. Moreover, this application emphasizes the problem of accessibility to a facility from the container's entry. This problem has been resolved by inserting the user's knowledge in the problem description. Here, the designer decided to insert a free corridor in the shelter in order to keep a space of accessibility to all the components.

However, when the designer decides to insert his job knowledge in the problem formulation, he automatically influences the search of solutions realised by the algorithm. If he wants to find innovative solutions, he has to simplify the problem description and to adapt the design constraints and the objectives to his design preferences. For example, in the application studied in this paper, the idea is to consider the accessibility to the facilities as a design constraint or an objective. Next research works have to explore this new concept.

Actually, this innovative optimization process proposed in this paper also suggests that the method could be improved according to the designer preferences:

- Qualitative fitness could be inserted into design objectives. When solving complex design problems, as a layout problem for example, the translation of some constraints and objectives into simple mathematical expressions can be very difficult. It means that these constraints and objectives could be replaced by a mark given by designers in order to characterize their designs. Then, this qualitative fitness could be considered by the algorithm as a design objective. Alexandra Melike Brintrup has already developed an interactive genetic algorithm-based framework for handling qualitative criteria in design optimization [14]. Actually, it should improve the performances of Pareto-optimal solutions.
- The designer could interact with design variables during the optimization process. Stopping the optimizer would allow the designer to firstly analyze a specific solution, secondly locally modify the design configuration and then decide to keep this modified design in the next

generation of the genetic algorithm. We can find in [15] a significant contribution to this concept applied to the design optimization of architectural layouts.

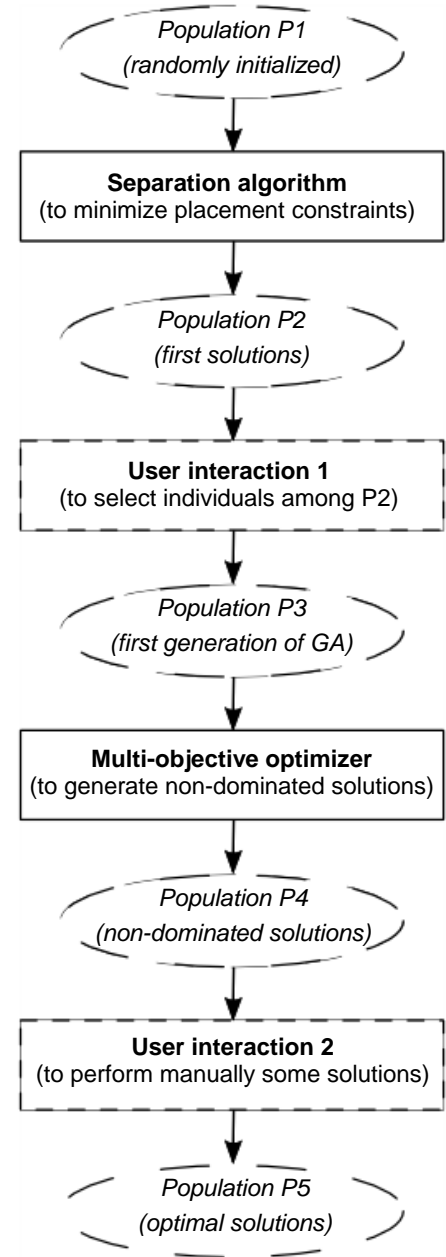


Figure 8: Schematic representation of the optimization strategy.

5 CONCLUSION

This article presents an innovative layout problem formulation including the concept of space of accessibility defined in section 3. It shows that problem formulation is a very important step in the optimization process because it has a great impact on computed solutions. Secondly, the hybridization of the separation algorithm and the multi-objective algorithm is a very efficient method to ensure a good diversity in a Pareto-optimal solutions set. Moreover, the strategy is designed to allow the interaction between the user and the optimization process in order to improve the performances of Pareto-optimal designs.

Actually, for industrial experts, design optimization has great advantages. On the one hand, it allows the designers to explore more alternative solutions to their problem. This is a very good way to encourage the innovation. On the other hand, using design optimization

lets the designer to easily make his design choices and justifying them with quantitative values related to his problem formulation.

Acknowledgement

The authors would like to acknowledge THALES for the application study.

6 REFERENCES

- [1] Giassi, A., Bennis, F., Maisonneuve, JJ., 2004, Multidisciplinary design optimization and robust design approaches applied to concurrent design, *Structural and Multidisciplinary Optimization*, 28: 356-371.
- [2] Dyckhoff, H., 1990, A typology of cutting and packing problems, *European Journal of Operational Research*, 44(2): 145-159
- [3] Drira, A., Pierreval, H., Hajri-Gabouj, S., 2007, Facility layout problems: A survey, *Annual Reviews in Control*, 31: 255-267
- [4] Cagan, J., Shimada, K., Yin, S., 2002, A survey of computational approaches to the three-dimensional layout problems, *Computer-Aided Design*, 34: 597-611
- [5] Yi, M., Fadel, GM., Gantovnik, VB., 2008, Vehicle configuration design with a packing genetic algorithm, *International Journal of Heavy Vehicle Systems*, 15: 433-448
- [6] Szyckman, S., Cagan, J., 1995, A simulated annealing-based approach to the three-dimensional packing, *Journal of Mechanical Design*, 117: 308-314
- [7] Szyckman, S., Cagan, J., 1997, Constrained three-dimensional component layout using simulated annealing, *Journal of Mechanical Design*, 119: 433-448
- [8] Su, Y., Cagan, J., 200, An extended pattern search algorithm for three-dimensional component layout, *Journal of Mechanical Design*, 122: 102-108
- [9] Imamichi, T., Nagamochi, H. 2007, A multi-sphere scheme for 2D and 3D packing problems, *SLS 2007, Proceedings of Engineering Stochastic Local Search Algorithms, Designing, Implementing and Analyzing Effective Heuristics, Lecture Notes in Computer Science*, 4638: 207-211
- [10] Imamichi T., Nagamochi, H., 2008, Designing algorithms with multi-sphere scheme, *ICKS 2008, Proceedings of the International Conference on Informatics Education and Research for Knowledge-Circulating Society*, 125-130
- [11] Poles, S., 2003, MOGA-II, an improved multi-objective genetic algorithm, *ESTECO, Technical Report 2003-2006*, 1-16
- [12] Deb, K., 1998, Multi-objective genetic algorithms: problems difficulties and construction of test problems, *Evolutionary Computation*, 7: 205-230
- [13] Serna, L., Mejia, R., Bennis, F., Fischer, X., 2005, Some ways of implementation of interactive design, *Virtual Concept*
- [14] Brintrup, AM., Ramsden, J., Tiwari, A., 2007, An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization, *Computers in Industry*, 58:279-291
- [15] Michalek, JJ., Papalambros, PY., 2002, Interactive design optimization of architectural layouts, *Engineering Optimization*, 34: 485-501