MODULAR OPTIMIZATION STRATEGY FOR LAYOUT PROBLEMS

Julien Bénabès[1], Emilie Poirson[1], Fouad Bennis[1] and Yannick Ravaut[2]
(1) Ecole Centrale de Nantes, IRCCyN UMR CNRS 6597, FR
(2) Thales Communications, FR

ABSTRACT
Layout design optimization has a significant impact in the design and use of many engineering products and systems. The search of an optimal layout configuration is a hard and critical task and designers need specific optimization strategies to achieve it. These strategies have to be generic to be adapted to a maximum of layout problems and a maximum of designer's requirements. These strategies have also to be effective in the search of optimal solutions without being too time consuming. Then, this paper proposes a modular optimization strategy based on the genetic algorithm. This strategy is adapted to multi-objective layout problems, when the designer has to make a final decision on optimal solutions. The optimization modules are used in order to make the search of feasible designs easier, by taking into account the specificities of layout problems. The different algorithms, which result from the modular optimization strategy, are tested and compared on a real-world application. This layout problem deals with the search of an optimal spatial arrangement of components inside a shelter.

Keywords: layout problems, design optimization, modular optimization strategy, 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. Layout problems are usually considered as optimization problems. Although there are, in layout design literature, different definitions of layout optimization problems [2], [3], the key idea is always the same: given a set of free form components and an available space, a layout problem consists of finding the best arrangement (location and orientation) of components satisfying geometrical and functional constraints and achieving design objectives. This generic definition can be adapted to all real-world applications. For example, Drira et al. [4] and Wäscher et al. [5] have adapted the definition of a layout problem to their respective research domain, meaning the facility layout design and the cutting and packing problems.

Cagan et al. proposed in [2] a schematic representation of the major constituent parts for solving a generic layout optimization problem. One finds in [6] another representation including three main parts: the description of the problem, the formulation of the problem and the optimization strategy. As it is shown in Figure 1, these three parts are connected.

In fact, the problem description defines the dimension of the layout problem (one, two or three dimensions) and describes the layout components, meaning the container and the components that have to be placed into the container. Then, this description and all the expert's requirements have to be translated into design variables, constraints and objectives in order to change the layout problem into an optimization problem. During the optimization process, the solving strategy continuously interacts with the formulation problem in order to compute the design constraints and the objectives. The optimization strategy can also include an interactive tool that allows the designer to interact with the process. This tool needs a graphical environment where the layout components are modeled according to the problem dimension.
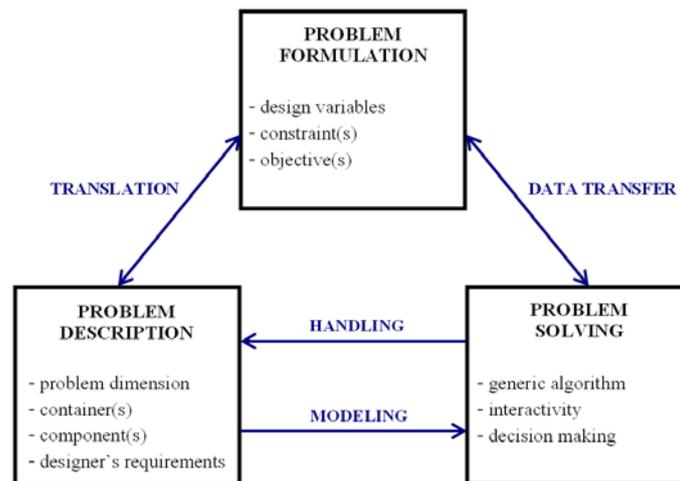
Figure 1. Schematic representation of a layout problem

This paper focus on a particular step of the layout optimization process: the problem solving. In fact, layout problems are generally considered as non-linear and NP-hard optimization problems. Problems are intrinsically harder than those which can be solved by a non-deterministic Turing machine in polynomial time. One finds multiple single or multi-objective approaches to solve layout optimization problems in two or three dimensions [2]. Traditional gradient-based approaches can be used for simple layout problems. For more complex real-world applications, some stochastic algorithms are required to avoid local optima. For example, some optimization strategies use genetic algorithms [3], simulated-annealing algorithms [7], [8] or extended pattern search algorithms [9]. Most search algorithms are developed for a specific problem and they provide an effective optimization strategy for it. However, they are not generic and can not be adapted to a lot of layout problems. Some of recent studies deal with the search of efficient generic algorithms for solving layout problems. Jacquenot et al. propose in [10] a hybrid algorithm based on a genetic algorithm coupled with a separation algorithm. A variant of this approach is also presented in [11].

In general, the development of an engineering object is considered as a single process involving multi-criteria identification of the mathematical model followed by multi-criteria optimization of the object design on the basis of this mathematical model. The direct participation of the designer in the construction of the feasible design and non-formal analysis are the essential stages of the search for the optimal design. For 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. One can find in [12] a significant contribution to this concept applied to the design optimization of architectural layouts. Moreover, interactivity with designer can be used to insert qualitative fitness or user perceptions in the design process [13]. In layout design, Brintrup et al. have already developed an interactive genetic algorithm based framework for handling qualitative criteria in design optimization [14].

Actually, this paper proposes a modular optimization strategy, based on the genetic algorithm. This optimization approach consists in combining a genetic algorithm with optimization modules, in order to help the genetic algorithm in the search of designs which respect the design constraints. This strategy is tested on the three dimensional layout problem of a shelter.

This paper is organized as follows: the section 2 presents the application studied in this paper. It presents the description and the formulation of the layout problem of a shelter. The section 3 explains why the genetic algorithm is used in the optimization approach proposed in this paper, by defining in particular the complexity of a layout problem. This section also presents the different algorithms which results from the modular approach. The section 4 is dedicated to comparison of the performances of the optimization algorithms, which result from the modular approach, on the layout problem of the shelter. The section 5 concludes this study.

2   LAYOUT PROBLEM OF A SHELTER

The application studied in this paper deals with the layout problem of a shelter. This problem is a three dimensional layout problem. Eight components have to be located in the shelter, including four

electrical and energy cabinets, two desks and two electrical boxes. The CAD model of the shelter is presented in Figure 2.
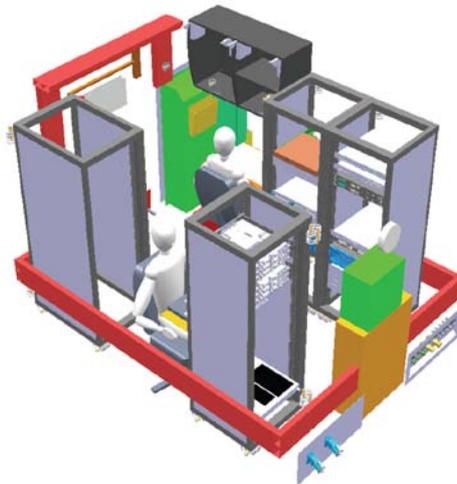

Figure 2. 3D CAD model of the shelter

This problem has been simplified in two dimensions, according to the fact that the cabinets are full height of the shelter and prevent a superposition of components. One of the possible designs of the shelter, modeled in two dimensions, is presented in Figure 3. The description and the formulation of this model are also described in [6] and [11].
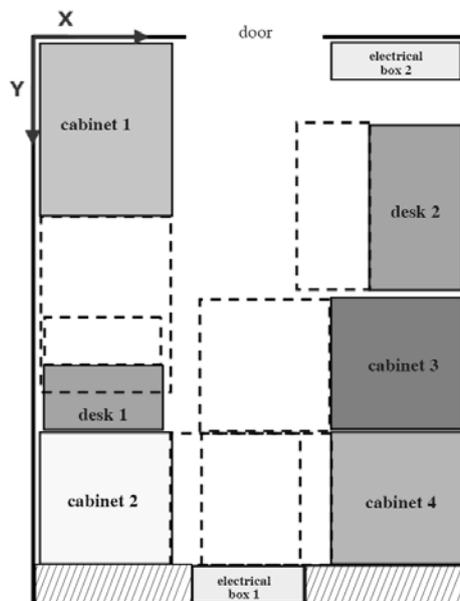

Figure 3. Description of the shelter in two dimensions

The layout components are divided in two categories of components: the material and the virtual components. A material component has a mass and can not overlap with another material component. On the other hand, a virtual component has no mass and can overlap with a material or a virtual component, according to the designer's requirements.

In fact, in the layout problem of the shelter, the layout components are made up of:
- 8 material components: 4 cabinets, 2 desks and 2 electrical boxes
- 6 spaces of accessibility (dotted rectangles in Figure 3)
- 1 free space (hatched rectangle in Figure 3) located below the air-conditioner, where no cabinet can be placed
- 1 free space in front of the door
- 1 free corridor located in the middle of the shelter

A space of accessibility is linked to a material component. For example, the space of accessibility of the cabinet 1 is defined as the required space, placed in front of the cabinet 1, and used to insert some materials into the cabinet 1. No material component can be placed in the space of accessibility.

The free corridor is used to guarantee that all components are accessible from the shelter's entry. This free corridor, the other free spaces and the spaces of accessibility can be considered as virtual components.

The density of this layout problem, defined as the ratio between the space occupied by the components and the available space in the container, is equal to 105%. This density does not take into account that a virtual component can overlap with another virtual component.

The placement of each material component is defined by two continued variables (X, Y) for the position, one discrete variable (α) for the orientation and one discrete variable (λ) for the direction.

The layout problem of this shelter is a multi-constraints and multi-objectives optimization problem. Four non-overlap constraints $(C1, C2, C3, C4)$ are defined:

- C1: non-overlap constraints between material components
- C2: non-overlap constraints between material and virtual components
- C3: non-overlap constraints between components and the exterior of the shelter
- C4: non-overlap constraints between the cabinets and the free space located below the air-conditioner

Because of the dimension of this problem and the rectangular shape of the layout components, the overlap is defined as the intersection area between components. The intersection area between the components i and j is defined by:

$$A_{ij} = \max[0, \min(x_i + \frac{l_i}{2}) - \max(x_j - \frac{l_j}{2})] \times \max[0, \min(y_i + \frac{L_i}{2}) - \max(y_j - \frac{L_j}{2})] \qquad (1)$$

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

Two objectives $(\min O1, \max O2)$ are also defined. The objective $O1$ is used to balance the masses inside the shelter. It means to minimize the distance between the center of gravity of the layout components and the geometrical center of the shelter. The objective $O2$ is used to maximize the distance between the cabinet 1 (energy network) and the cabinets 2 and 3 and the electrical box 2 (electrical network). All the distances between components, used in the computation of the two objectives, are the distances between the geometrical centers of components defined by the Euclidean norm. More information about the definition of these objectives is available in [11].

3 MODULAR OPTIMIZATION STRATEGY

The principle of this modular optimization strategy is to use a genetic algorithm and to insert in this algorithm some optimization modules in order to improve the global performances of the optimization strategy. This section presents the different algorithms which result from this modular approach.

3.1 Genetic algorithm

This paper proposes to use the genetic algorithm process because of the great complexity of real-world layout problems. The complexity of a layout problem is defined in [6]. This complexity is linked to the geometry of layout components, the layout density and the problem formulation.

The problem formulation has a significant impact on the choice of an appropriate optimization algorithm. More numerous the constraints and objectives are, more complex the search of a feasible design is. The design space is parceled and the designer can not use traditional gradient-based optimization approaches to pass to a feasible region to another one. Stochastic algorithms, as the genetic algorithm, have to be used and the calculation time increases so that the problem is more complex.

In this paper, the Genetic Algorithm Omni-Optimizer is used [15]. This algorithm is designed to handle single and multi-objective problems. Given a set of initial individuals, randomly generated, the genetic algorithm uses basically three operators in order to create a set of new solutions. These genetic operators are the selection, the crossover and the mutation.
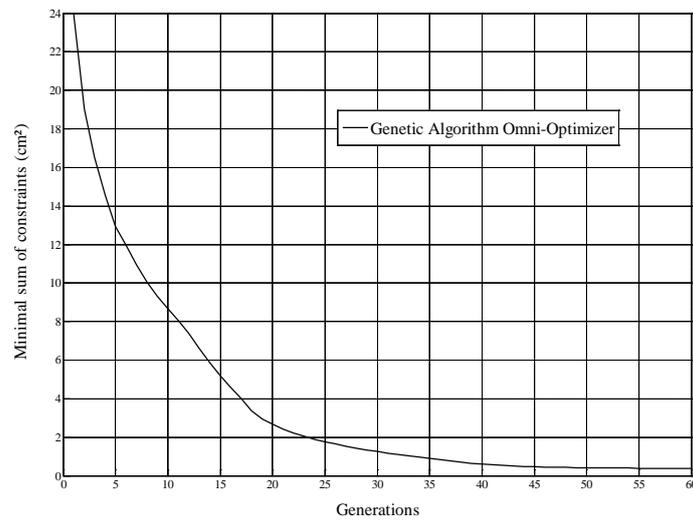
Figure 4. Convergence of the genetic algorithm

Figure 4 illustrates the convergence of the Genetic Algorithm Omni-Optimizer for the layout problem of the shelter. The axis of ordinates represents, for each generation of the genetic algorithm, the minimal sum of constraints obtained by one design. It means that, for a specific generation, if this minimal sum of constraints is equal to 0, the genetic algorithm has found a "feasible" design, meaning a design which respects all the non-overlap constraints. This algorithm has been initialized by 200 designs, randomly created. Figure 4 shows that the genetic algorithm converges after 60 generations but it can not find a feasible solution, which respects all the placement constraints. The layout problem of the shelter is too complex for this genetic algorithm. Consequently, the main objective of the following sub-sections is to describe some optimization modules, which can be linked to the genetic algorithm in order to help it in the search of feasible designs. These optimization modules are specific to layout optimization problems.

3.2 Module 1: "Optimization of the direction of the components"
In lots of real-world applications, as the layout problem of the shelter, one can find material and virtual components. This optimization module is specific to these layout problems, which deals with the spatial arrangement of material and virtual components.

Let us consider the application studied in this paper. The shelter is modeled in two dimensions and all the layout components have a rectangular shape. Let us consider a material component 1 (for example the cabinet 1) and a material component 2 (for example the electrical box 2). A virtual component (for example the space of accessibility of the cabinet 1), represented by a dotted rectangle in Figure 5, is linked to the component 1. Figure 5 illustrates the two possible directions (λ) for the component 1, meaning the two possible positions of the virtual component.
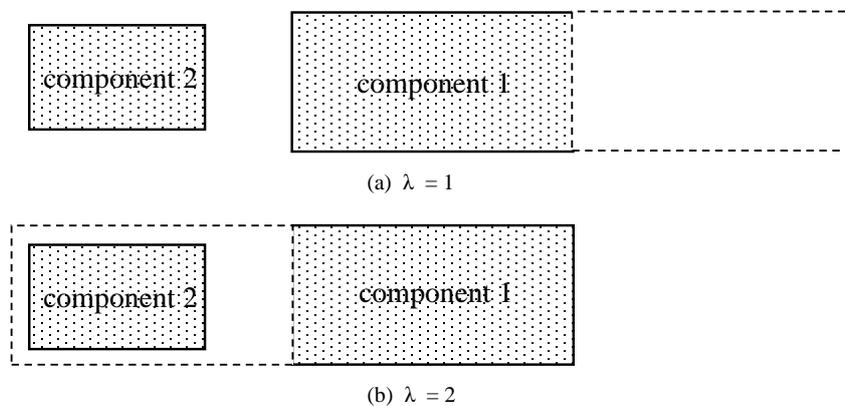

(a) $\lambda = 1$

(b) $\lambda = 2$

Figure 5. Direction of a layout component

The objective of the module 1 "Optimization of the direction of the components" is to optimize the placement of this virtual component, by deleting the discrete variable λ used for the direction and minimizing the non-overlap constraints betweens virtual components and material components. For example, Figure 5 shows that the module 1 chooses the case (a), with a direction equal to 1 because this direction minimizes the non-overlap constraint between the space of accessibility of the component 1 and the component 2.

3.3 Module 2: "Separation algorithm"
Given a layout configuration that does not satisfy placement constraints, the objective of the separation algorithm is to minimize the non-respect of overlap and protrusion constraints. In fact, the separation algorithm is designed to solve a single objective optimization problem, where all the placement constraints are gathered into one objective function.

This separation algorithm has been tested on a simple two dimensional layout problem, which deals with the search of an optimal spatial arrangement of square components inside a square container. Different optimization problems with different densities have been tested and the results are presented in [10].

3.4 Module 3: "Local disruption"
This module is used to help the separation algorithm to find a feasible design, meaning a design which respects all the placement constraints. In fact, the separation algorithm, proposed in this paper, only modifies the position variables, it means the continued variables. It has not effect on the orientation or direction variables. Consequently, for some layout configurations, the separation algorithm does not find a feasible solution. It finds a local minimum which does not respect all the placement constraints. One or several components have to be rotated.

The main idea of this module 3 is to associate to the separation algorithm a local disruption in the layout configuration. This local disruption is used to randomly change the orientation of some components. The components which are rotated are randomly selected. The number of rotated components is fixed by the designer. For example, it can be fixed to 30% of the total number of components of the layout problem.

In fact, a local "disruption" already exists in the genetic algorithm process. This disruption is realized by the genetic operator "mutation". However, the mutation is not dedicated to specific optimization variables. The module 3 is dedicated to the variables which define the orientation of components. Consequently, this optimization module is more effective in the search of feasible designs.

3.5 Modular optimization strategy
This sub-section describes the different algorithms which results from the modular optimization approach proposed in this paper. Figure 6 illustrates two optimization algorithms, created by combining:
- Figure 6 (a) : the genetic algorithm with the modules 1 and 2
- Figure 6 (b) : the genetic algorithm with the modules 1, 2 and 3

The module 3 can be repeated j_{max} times, where j_{max} is fixed by the designer. Between two applications of the module 3, the optimization process tests if the current design respects all the placement constraints. If so, the optimization process evaluates the design and moves to the next solution according to the genetic operators. Otherwise, the module 3 is run if $j < j_{max}$.

Moreover, if $j = j_{max}$ and if the separation algorithm has not found a feasible solution, the optimization process continues by considering the design that minimizes the placement constraints among all the designs generated by the separation algorithm.

Moreover, a hybrid algorithm, used for solving layout problems, is described in [9]. This algorithm is obtained by combining the genetic algorithm with the optimization module 2.
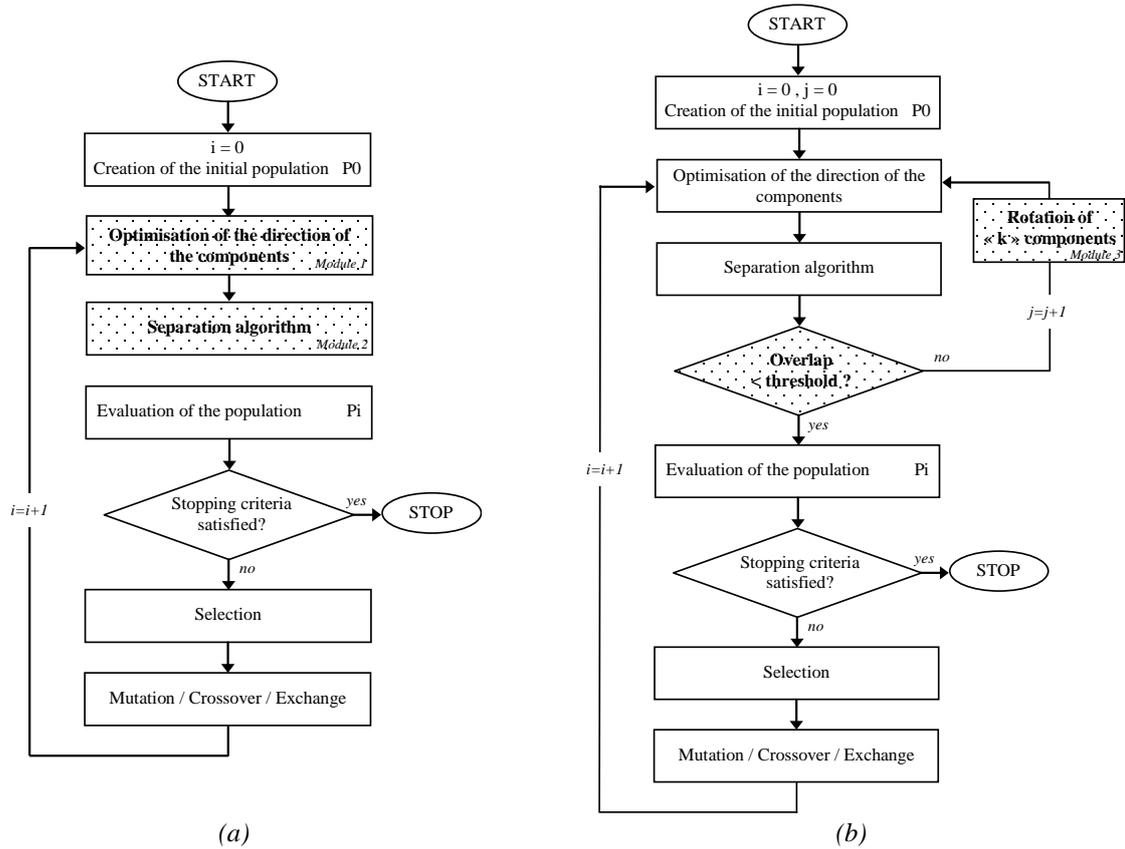
(a)   *(b)*

Figure 6. Modular optimization strategy

4   COMPARISON OF THE OPTIMIZATION STRATEGIES

This section presents the results obtained on the real-world layout problem of the shelter by considering the different optimization algorithms which result from the modular optimization approach.

In order to compare the performances of the different algorithms, the number of feasible variants generated by each algorithm is measured for each generation. In fact, the design j is a new variant 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 and Y (Δ is fixed to 500mm in the following simulations)
- one of the components has been rotated
- the minimum difference between the objective values of the two designs is bigger than a limit, fixed to 10 cm

This number of variants is plotted according to the calculation time. By associating the different modules with the genetic algorithm, with different parameters, six optimization algorithms are created:

- optimization algorithm A: genetic algorithm with the module 1
- optimization algorithm B: genetic algorithm with the module 2
- optimization algorithm C: genetic algorithm with the modules 1 and 2
- optimization algorithm D: genetic algorithm with the modules 1, 2 and 3, with $j_{max} = 3$
- optimization algorithm E: genetic algorithm with the modules 1, 2 and 3, with $j_{max} = 10$
- optimization algorithm F: genetic algorithm with the modules 1, 2 and 3, with $j_{max} = 20$

The parameters of the genetic algorithm Omni-Optimizer are the same for each simulation. The size of the population is fixed to 200 individuals. The maximal number of iterations is fixed to 100.

Firstly, let us consider the comparison of the optimization algorithms A, B and C. Figure 7 illustrates the results obtained for these three algorithms.
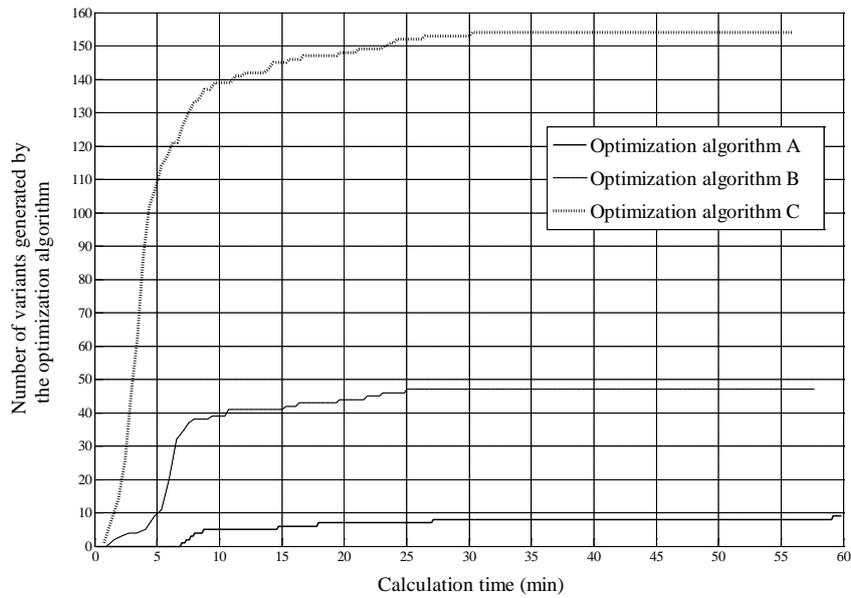
Figure 7. Comparison of the algorithms A, B and C

Firstly, Figure 7 shows that all the optimization algorithms have converged because the algorithms are not able to find more variants. For each algorithm, the maximal number of iterations has also been attained on the period of calculation time defined by the graph.

Secondly, the results underline the great influence of the module 1 on the performances of the algorithms. In fact, the algorithm A has generated 9 variants whereas the genetic algorithm without optimization modules was not able to find feasible solutions. By the same way, the results also underline the great influence of the module 2 and the combination of the two optimization modules 1 and 2. The optimization algorithm C found more variants than the algorithm B, which found more variants than the algorithm A.

Moreover, Figure 7 shows that the optimization modules 1 and 2 allow the algorithm to generate variants more quickly.

On the other hand, it can be relevant to evaluate the different Pareto-optimal variants, generated by these three algorithms, according to their objective values. Let us plot a scatter graph with all the Pareto-optimal solutions found by the three algorithms. Figure 8 illustrates this scatter graph.
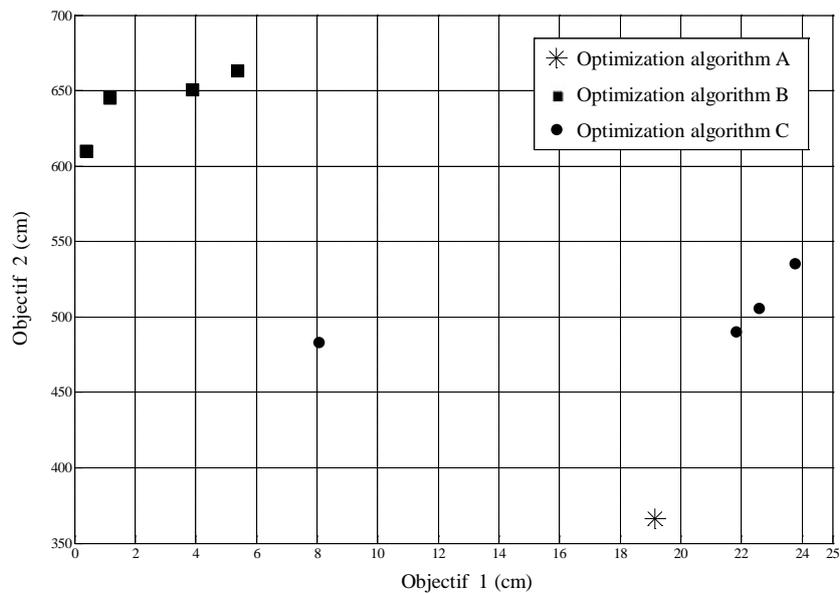

Figure 8. Pareto-optimal variants for the algorithms A, B and C

Figure 8 shows that the Pareto-optimal solutions generated by the algorithm C have better performances than the other solutions. In summary, the optimization algorithm C, based on a genetic algorithm combined with the modules 1 and 2 has the best performances.

Let us now consider the algorithms C, D, E and F. By the same way as Figure 7, Figure 9 illustrates the results obtained by these four algorithms.
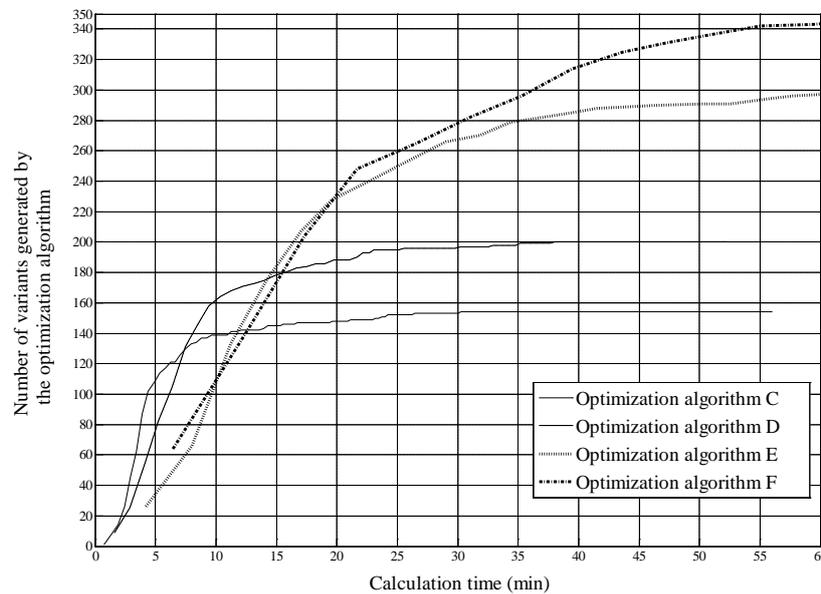

Figure 9. Comparison of the algorithms C, D, E, and F

Firstly, in this simulation, on the period of calculation time defined by the graph, the maximal number of iterations is only attained for the algorithms C and D. In fact, the algorithms E and F use the optimization module 3, with the parameter j_{max} respectively equal to 10 and 20, and this optimization module is time consuming. When the module 3 is run, the module 1 and 2 are also run. Each generation of the algorithm takes more time to generate designs and the maximal number of generations is not attained on the period of calculation time defined by the graph.

Secondly, it is difficult here to define which optimization algorithm has the best performances because it depends on the designer's requirements. The optimization algorithms can be sorted by ascend with the number of generated variants in the order C, D, E and F. It means that the module 3 has a great influence on the search of feasible variants because the local disruption helps the separation algorithm in the search of feasible designs.

However, the algorithm C finds variants before the other optimization algorithms. Consequently, the choice of the optimization algorithm depends on the designer's requirements, if he wants to find lots of solutions without caring about the calculation time or if he wants to generate few solutions in a short period of calculation time. Moreover, the Pareto-optimal variants generated by the optimization algorithms have similar performances.

5   CONCLUSION

Layout problems are generally considered as complex problems and stochastic optimization algorithms have to be used. The complexity of layout problems is usually linked to the problem formulation. The design space is parcelled so that the designer needs specific optimization algorithms as the genetic algorithm in order to find feasible solutions.

However, the application studied in this paper shows that the genetic algorithm is not completely adapted to very complex layout problems. The layout problem of the shelter deals with the search of an optimal spatial arrangement of material and virtual components. This problem has a big density and its formulation is complex.

Consequently, this paper proposes a modular optimization approach, based on the genetic algorithm, in order to help the algorithm in the search of feasible designs. Three optimization modules are described: the optimization of the direction of the components, the separation algorithm and the local disruption. The designer can decide to insert these modules in the genetic algorithm process, according

to the specificities of his layout problem. For example, the module 1 is only dedicated to layout problems which deal with the spatial arrangement of material and virtual components.

The results described in this paper show that the optimization modules enable the designer to generate more feasible variants, meaning more designs which respect all the placement constraints. The optimization modules are time consuming. The designer has to choose the most appropriate module according to his requirements: to quickly generate variants with average performances or to generate lots of variants with good performances. In the two scenarios, the calculation time is not the same.

The development of this modular optimization approach is part a global design approach used to solve complex layout problems. These optimization algorithms are dedicated to multi-objective optimization problems. The number of variants, generated by these algorithms is very big. However, the designer has to make a final choice, by choosing one of these optimal variants. Outlooks are now dedicated of the development of methods and tools which enable the designer to make this final choice. These tools provide interaction between the designer and the optimization process, in order to locally modify a design, modify the problem formulation or insert the personal judgment of the designer in the search of optimal solutions.

ACKNOWLEDGMENT

REFERENCES

[1] Giassi, A., Bennis, F., Maisonneuve, J.-J., Multidisciplinary design optimization and robust design approaches applied to concurrent design. *Structural and Multidisciplinary Optimization*, 2004, 28, 356–371

[2] Cagan, J., Shimada, S., Yin, S., A survey of computational approaches to the three-dimensional layout problems. *Computer-Aided Design*, 2002, 34, 597–611

[3] Yi, M., Fadel, G., Gantovnik, V.B., Vehicle configuration design with a packing genetic algorithm. *International Journal of Heavy Vehicle Systems*, 2008, 15, 433–448

[4] Drira, A., Pierreval, H., Hajri-Gabouj, S., Facility layout problems: A survey. *Annual Reviews in Control*, 2007, 31, 255–267

[5] Wäscher, G., Haubner, H., Schumann, H., An improved typology of cutting and packing problems. *European Journal of Operational Research*, 2007, 183, 1109–1130

[6] Bénabès, J., Bennis, F., Poirson, E., Ravaut, Y., A new approach for specifying and solving layout problems. In *proceedings of the IDMME-Virtual Concept*, 2010, Bordeaux (France).

[7] Szykman, S., Cagan, J., Constrained three-dimensional component layout using simulated annealing, *Journal of Mechanical Design*, 1997, 119, 28–35

[8] Shen, H., Subic, A., Bourmistrov, A., Three-dimensional design layout optimization for mechatronic devices using the simulated annealing approach. In *proceedings of the 15th International Conference on Engineering Design*, 2005, Melbourne (Australia)

[9] Su, Y., Cagan, J., An extended pattern search algorithm for three dimensional component layout, *Journal of Mechanical Design*, 2000, 122, 102–108

[10] Jacquenot, G., Bennis, F., Maisonneuve, J.-J., Wenger, P., 2d multi-objective placement algorithm for free-form components. In *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*, 2009, San Diego (USA)

[11] Bénabès, J., Bennis, F., Poirson, E., Ravaut, Y., Interactive optimization strategies for layout problems. *International Journal on Interactive Design and Manufacturing*, 2010, 4, 181–190

[12] Michalek, J.-J., Papalambros, P.-Y., Interactive design optimization of architectural layouts. *Engineering Optimization,* 2002, 4, 485–501

[13] Poirson, E., Petiot, J.-F., Gilbert, J., Integration of user perceptions in the design process: application to musical instrument optimization. *Journal of Mechanical Design*, 2007, 129, 1206–1214

[14] Brintrup, A.-M., Ramsden, J., Tiwari, A., An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization. *Computers in Industry,* 2007, 58, 279–291

[15] Deb, K., Tiwari, S., Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 2008, 185, 1062–1087

Contact: Julien Bénabès
PhD Student
Ecole Centrale de Nantes, IRCCyN, UMR CNRS 6597
1 rue de la Noë, BP 92101
44321 Nantes
France
Phone: +33 2.40.37.69.42
Fax: +33 2.40.37.69.30
E-mail Address: julien.benabes@irccyn.ec-nantes.fr