Interactive modular optimization strategy for layout problems

Julien Bénabès, Emilie Poirson, Fouad Bennis, Yannick Ravaut

Proceedings of the ASME 2011 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2011
August 29-31, 2011, Washington, DC, USA

DETC2011-47925

INTERACTIVE MODULAR OPTIMIZATION STRATEGY FOR LAYOUT PROBLEMS

Julien Bénabès
PhD Student
Ecole Centrale de Nantes
IRCCyN
Nantes, France
Email: julien.benabes@irccyn.ec-nantes.fr

Emilie Poirson
Associate Professor
Ecole Centrale de Nantes
IRCCyN
Nantes, France
Email: emilie.poirson@irccyn.ec-nantes.fr

Fouad Bennis
Professor
Ecole Centrale de Nantes
IRCCyN
Nantes, France
Email: fouad.bennis@irccyn.ec-nantes.fr

Yannick Ravaut
Engineer Thales Communications
Cholet, France
Email: yannick.ravaut@fr.thalesgroup.com

ABSTRACT

Layout design optimization has a significant impact in the design and use of many engineering products and systems. Real-world layout problems are usually considered as complex problems because of the geometry of components, the problem density and the great number of designer's requirements. Solving these optimization problems is a hard and time consuming task. This paper proposes an interactive modular optimization strategy which allows the designer to find optimal solutions in a short period of calculation time. This generic strategy is based on a genetic algorithm, combined with specific optimization modules. These modules improve the global performances of the algorithm. This approach is adapted to multi-objective optimization problems and interactivity between the designer and the optimization process is used to make a final choice among design alternatives. This optimization strategy is tested on a real-world application which deals with the search of an optimal spatial arrangement of a shelter.

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 defined as optimization problems and in layout design literature, one finds some different definitions of layout optimization problems [2, 3]. However, 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 i.e. facility layout design and cutting and packing problems.

A global layout optimization process can be divided in three main steps : the description of the problem, the formulation of the problem and the optimization strategy. The problem description defines the dimension of the layout problem and identifies the layout components, meaning the container and the components which have to be placed into the container. Then, this description and all the designer's requirements have to be translated into design variables, constraints and objectives in order to change the layout problem into an optimization problem.

This paper focuses on the third step, the optimization strategy, which deals with the search of optimal designs with respect all the designer's requirements. 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 [6] or extended pattern search algorithms [7]. 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 [8] an hybrid algorithm based on a genetic algorithm coupled with a separation algorithm. A variant of this approach is also presented in [9].

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. We can find in [10] 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 [11]. In layout design, Brintrup et al. have already developed an interactive genetic algorithm based framework for handling qualitative criteria in design optimization [12]. Also, the designer can interact with the optimization process in order to make a final choice on the alternatives proposed by the optimization algorithm. Interactive decision making environments are necessary to make this final choice [13, 14]

Actually, this paper proposes an interactive modular optimization strategy which allows the designer to solve complex and multi-objective layout optimization problems. The modular approach is based on a genetic algorithm and uses specific optimization modules in order to help the optimization algorithm to find more design alternatives in a short period of calculation time. In this paper, a multi-objective optimization approach is used. It means that the strategy, proposed in this paper, provides some interactive tools to allow the designer to make a final choice on design alternatives. The designer can interact with a solution in order to take into account his personal judgment in the choice of an optimal design. This optimization approach is tested on a real-world layout problem. This problem deals with the optimal spatial arrangement of components inside a shelter.

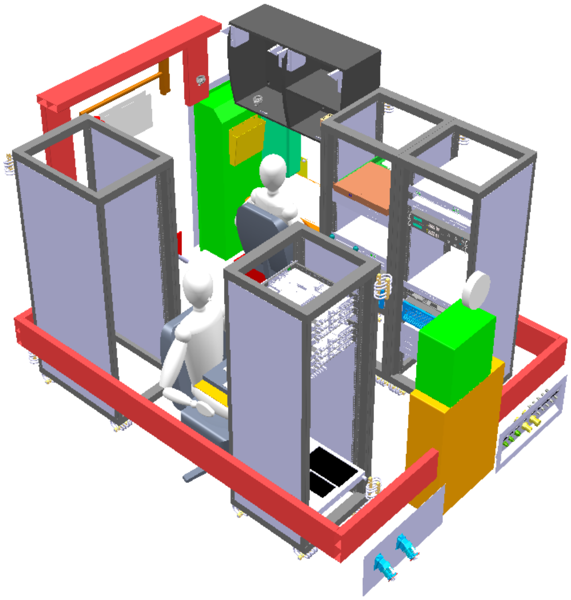


FIGURE 1. OVERALL VIEW OF THE SHELTER.

This paper is organized as follows: the section 2 presents the real-world application studied in this paper. It describes in particular the description and the formulation of the layout problem. The section 3 is dedicated to the modular optimization strategy. The different algorithms which result from this modular approach are compared on the layout problem of the shelter. Then, the following section describes the tools, provided by the optimization strategy proposed in this paper, which allow the designer to make a final choice on optimal design alternatives. Actually, the last section concludes this study.

LAYOUT PROBLEM OF A SHELTER

The application studied in this paper deals with the layout problem of a shelter. Eight components have to be located in the shelter, including electrical and energy cabinets, desks and electrical boxes. The CAD model of the shelter is presented in Fig. 1.

This problem is a two dimensional layout optimization problem, according to the fact that the cabinets are full height of the

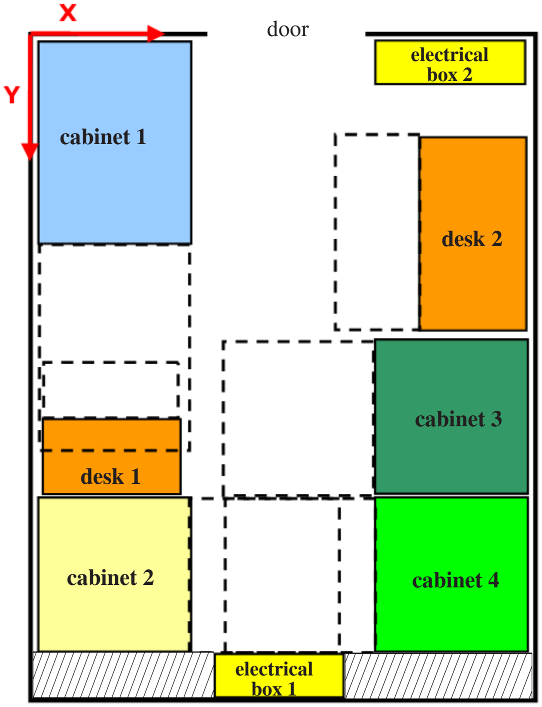


FIGURE 2. DESCRIPTION OF THE SHELTER IN 2D.

shelter and prevent a superposition of components. One of the possible designs of the shelter, modeled in two dimensions, is presented in Fig. 2. The description and the formulation of this model are also described in [9] and [15].

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 Fig. 2)
- 1 free space (hatched rectangle in Fig. 2) 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 a 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 of the geometrical center of the component, one discrete variable (α) for the orientation and one discrete variable (λ) for the direction. As all the layout components are modeled as rectangles in two dimensions, we consider that the orientation (α) can take two values : 0° or 90°. The direction (λ) can also take two values (1 or 2) which define the position of the space of accessibility in relation to the material component.

The layout problem of this shelter is a multi-constraints and multi-objectives optimization problem. Four non-overlap constraints are defined:

- C1: non-overlap constraints between material components
- C2: non-overlap constraints between material components 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 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:

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

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 3 and 4 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 are available in [9].

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.

Genetic Algorithm

This paper proposes to use a genetic algorithm process because of the great complexity of real-world layout problems. The complexity of a layout problem is defined in [15]. 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, more complex the search of a feasible design. 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 [16]. 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. This genetic algorithm has been tested on the layout problem of the shelter. The algorithm has been initialized by 200 designs, randomly created. The maximal number of iterations is fixed to 100. The simulation has been run ten times because of the stochastic behavior of the genetic algorithm. The convergence of the genetic algorithm Omni-Optimizer is illustrated in Fig. 3. In Fig. 3, 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. Then, the results, described in Fig. 3, show that the genetic algorithm converges but 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 idea of the modular strategy is to combine the genetic algorithm with some specific optimization modules, in order to help the generic algorithm to find more feasible design alternatives. These modules are described in the following sub-section.

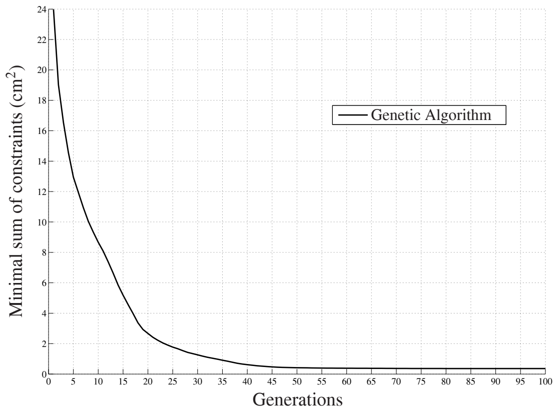
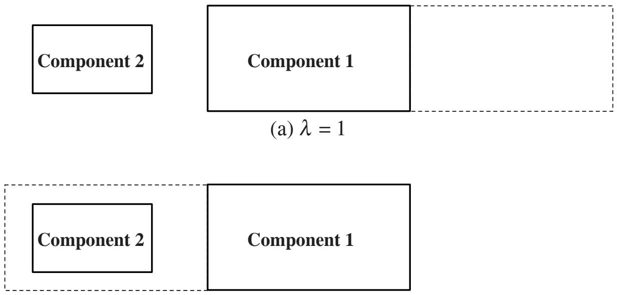


FIGURE 3. CONVERGENCE OF THE GENETIC ALGORITHM.



(a) λ = 1

(b) λ = 2

FIGURE 4. DIRECTION OF A LAYOUT COMPONENT.

Optimization modules

The objective of these optimization modules is to help the genetic algorithm to find more feasible designs. In this paper, three optimization modules are presented.

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 a layout problem, modeled in two dimensions, where the container and all the layout components have a rectangular shape. The problem can be for example the layout problem of the shelter. 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 Fig. 4, is linked to the component 1. Figure 4 illustrates the two possible directions (λ) for the component 1, meaning the two possible positions of the virtual 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 between virtual components and material components. For example, Fig. 4 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.

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 [9].

A hybrid optimization algorithm is presented in [8]. It combines the genetic algorithm Omni-Optimizer with the optimization module 2 “Separation Algorithm”.

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 X and Y. It has no 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.

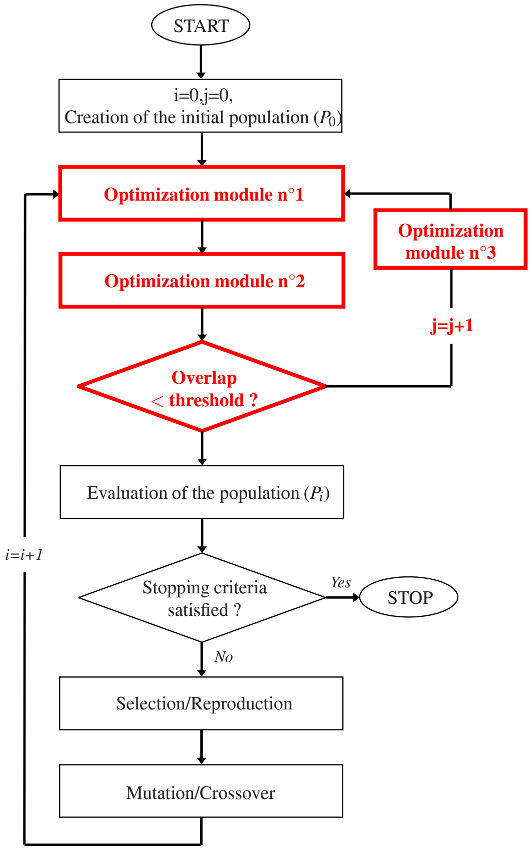


FIGURE 5. MODULAR OPTIMIZATION STRATEGY.

Modular optimization strategy

Different optimization algorithms result from this modular approach. It depends on the layout problem and the designer’s requirements. Figure 5 illustrates the optimization algorithm which results from the combination of the genetic algorithm Omni-Optimizer and the optimization 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.

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 variants generated by each algorithm is measured for each generation. In fact, the design i is a new variant if it differs from the design j 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 and (Δ is fixed to 50 cm 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 in the following simulations

This number of feasible variants, which respect all the placement constraints, is plotted according to the calculation time. By associating the different modules with the genetic algorithm, with different parameters, five optimization algorithms are proposed:

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

The initial population and 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. For each optimization algorithm, the simulation has been run ten times because of the stochastic behavior of the algorithm. Figure 6 illustrates the results obtained by the different optimization algorithms tested on the layout problem of the shelter.

Firstly, in this simulation, on the period time defined by the graph, the maximal number of iterations is only attained for the algorithms A, B and C. In fact, the algorithms E and F use the optimization module 3, 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 global 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, according to the number of generated feasible variants, in the order A, B, C, D, E. It means that the optimization modules 1, 2 and 3 have a significant influence on the search of variants, because it helps the genetic algorithm in the search of feasible designs.

Consequently, the choice of an approriate optimization algorithm depends on the layout optimization problem and on the designer's requirements. It is important to know if the designer 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.

It can also be relevant to evaluate the Pareto-optimal variants, generated by these optimization algorithms, on this period of calculation time. If we compare their objective values, we can notice that the Pareto-optimal designs, found by the algorithms B, C, D and E have similar performances. Their performances are better than the performances of the Pareto-optimal designs generated by the algorithm A.

INTERACTIVE DECISION MAKING

In 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. The Pareto-front is the set of Pareto-optimal solutions, meaning solutions which are not dominated by other solutions. We consider that a design U dominates a design V (Pareto dominance) if U is as good as V for all the objectives and better than V for at least one objective. Mathematically, this can be formulated by:

$$\begin{cases} \forall\, i \in [1,...,n]\; f_{iU} \leq f_{iV} \\ \exists\, j \in [1,...,n]\; f_{jU} < f_{jV} \end{cases} \tag{2}$$

where the design U is represented by a vector of objectives values $F_U = (f_{1U}, f_{2U}, ..., f_{mU})$, where f_{iU} is the i^{th} component of the vector of objectives F for the solution U.

Detection of variants

Let us consider the set of Pareto-optimal solutions generated by the optimization algorithm B. This set of designs is made up of 272 layout designs. It is too difficult for the designer to make a final decision on this big number of solutions. Because some Pareto-optimal solutions are very close, the approach, proposed in this paper, suggests to detect only the variants generated by the algorithm. A variant is defined in the previous section.

Let us consider different values of the parameter Δ and the number of detected variants for each value. The results are described in Tab. 1. For the results described in the next subsection, Δ is fixed to 1 m.

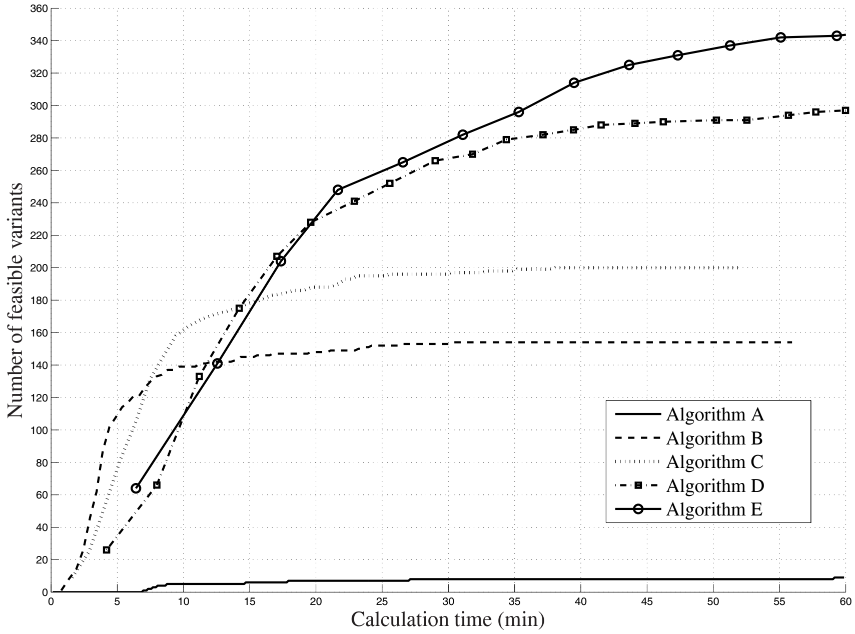


FIGURE 6. COMPARISON OF THE OPTIMIZATION ALGORITHMS.

TABLE 1. EVOLUTION OF THE NUMBER OF VARIANTS ACCORDING TO Δ.

Δ (mm)	Variants	Pareto-optimal variants
0	17334	272
50	904	21
500	88	8
1000	64	7

Interactive environment

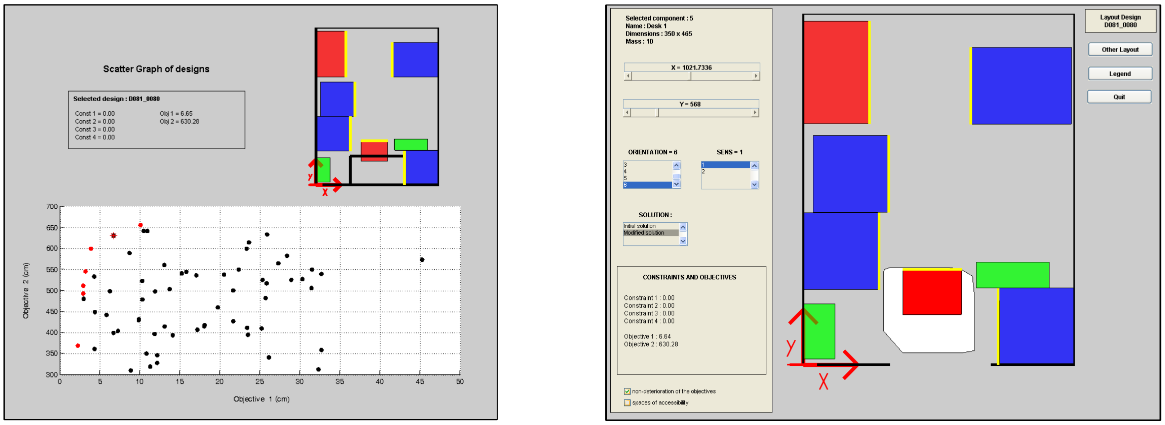
In most real-world layout problems, all designer's requirements can not be integrated in the layout problem formulation by adding simple mathematical expressions. Then, when the designer is exploring the set of Pareto-optimal solutions, he has to make a final decision on optimal designs according to his personal judgment.

Consequently, this paper proposes an interactive environment for decision making, in order to allow the designer to:

- explore the set of Pareto-optimal variants
- visualize the solution in two or three dimensions
- manually and locally modify the solution by changing the position of some components and by visualizing the new values of the constraints and the objectives that result from these modifications.

The main objective of this interactivity with the designer is to improve the performances of the optimal solutions proposed by the algorithm, by inserting in the decision making the personal judgment of the designer. This interactivity needs a efficient graphical and numerical environment, as shown in the Fig. 7.

Figure 7(a) illustrates the interactive tool which allows the designer to explore the set of Pareto-optimal solutions. The designer can click on a point on the graph and visualize the layout design, the constraints and the objectives related to this solution. Figure 7(b) illustrates the interactive tool which allows the designer to locally modify a selected design and compare the modified solution with the initial one and the solution initially proposed by the engineering experts. In this figure, we can see that a white area is displayed around the layout component 5 (desk 1). The area represents the set of positions where the designer can place the layout component 5 without damaging the design objectives. It means that, considering the current orientation of the layout components, if the component 5 moves inside the white area, the new solution won't be dominated by the current one. For the designer, it is an indicator of non-deterioration of the constraints and objectives.



Selected design : D081_0080

Const 1 = 0.00
Const 2 = 0.00
Const 3 = 0.00
Const 4 = 0.00
Obj 1 = 6.65
Obj 2 = 630.28

(a) display of Pareto-optimal designs

Selected component : 5
Name : Desk 1
Dimensions : 350 x 465
Mass : 10

X = 1021.7336

Y = 568

Constraint 1 : 0.00
Constraint 2 : 0.00
Constraint 3 : 0.00
Constraint 4 : 0.00

Objective 1 : 6.64
Objective 2 : 630.28

(b) interactive display of a design

FIGURE 7. INTERACTIVE ENVIRONMENT FOR DECISION MAKING.

Results

Let us consider the results concerning the optimization algorithm B, previously described. According to Tab. 1, this algorithm, has generated 7 Pareto-optimal variants, with Δ equal to 1 m. As it is explained in the previous subsection, let us consider that the designer selects one of these Pareto-optimal designs. This solution is represented in Fig. 8(b). According to Tab. 2, this solution has better performances than the first solution initialy proposed by the expert engineers, and illustrated in Fig.8(a). It is important to mention that this initial solution was an intuitive design which had been generated only by considering geometric aspects.

Then the interactive envrionment suggests to the designer to localy and manualy change the position and the orientation of some components of the layout design, generated by the optimization algorithm. The solution, modified by the designer according to his own preferences, is illustrated in Fig. 8(c). Acoording to Tab. 2, this solution has as good performances as the solution proposed by the algorithm.

CONCLUSION

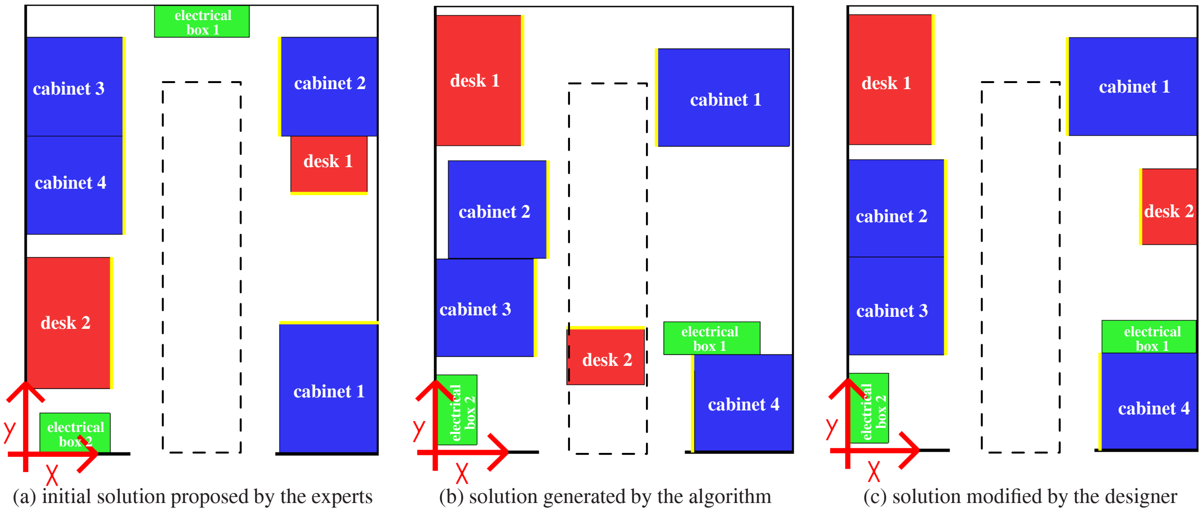
Layout problems are generally considered as complex problems and stochastic optimization algorithms have to be used. The complexity of layout problems is usually linked to the problem formulation. The design space is parcelled so that the designer needs stochastic optimization algorithms, as the genetic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

The application studied in this paper shows that the genetic algorithm is not completely adapted to very complex layout problems. In fact, the layout problem of a shelter, presented in this paper, 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, meaning four design constraints and two objectives. The genetic algorithm Omni-Optimizer does not find any solution for this optimization problem.

Consequently, this paper proposes a modular optimization approach, based on the combination of a genetic algorithm with specific optimization modules. These modules are used 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.

This modular optimization strategy has been tested on the layout problem of the shelter, studied in this paper. The results described in this paper show that the optimization modules enable the designer to generate more variants 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 lots of variants with average performances or to generate fewer variants with good performances. In the two cases, the calculation time is not the same.

The development of this modular optimization approach is a part of a global design approach used to solve complex layout problems. Because multi-objective optimization strategy is used in this approach, the algorithm generates lots of design alternatives. Consequently, the designer has to make a final choice, by choosing one of the optimal variants created by the optimization algorithm. Then, this paper proposes an interactive environment for decision making which allows the designer to detect the variants, visualize the optimal solutions on a scatter graph and locally modify a design according to his personal judgment. The main objective of this interactivity is to improve the global per-



(a) initial solution proposed by the experts (b) solution generated by the algorithm (c) solution modified by the designer

FIGURE 8. GRAPHICAL RESULTS.

TABLE 2. NUMERICAL RESULTS.

Objectives	initial solution (a)	selected design (b)	modified design (c)
Obj. 1 (cm,min)	25,41	6,65	8,33
Obj2. (cm,max)	604,88	630,28	648,43

formances of the optimal solution generated by the algorithm. Outlooks are now dedicated of the development of new interactive tools for decision making, dedicated to three-dimensional layout problems.

ACKNOWLEDGMENT

The authors would like to acknowledge Thales Communications for the application study.

References

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

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

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

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

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

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

[7] Su, Y., and Cagan, J., 2000. “An extended pattern search algorithm for threedimensional component layout”. *Journal of Mechanical Design,* ***122***, pp. 102–108.

[8] Jacquenot, G., Bennis, F., Maisonneuve, J.-J., and Wenger, P., 2009. “2d multi-objective placement algorithm for free-form components”. In Proceedings of ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2009.

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

[10] Michalek, J. J., and Papalambros, P. Y., 2002. “Interactive design optimization of architectural layouts”. *Engineering Optimization,* ***34***, pp. 485–501.

[11] Poirson, E., Petiot, J.-F., and Gilbert, J., 2007. “Integration of user perceptions in the design process: application to musical instrument optimization”. *Journal of Mechanical*

Design, ***129***, pp. 1206–1214.

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

[13] Balling, R., 1999. "Design by shopping: A new paradigm". In Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization, Buffalo, NY, pp. 295-297.

[14] Stump, G. M., Yukish, M., Simpson, T. W., and Harris, E. N., 2003. "Design space visualization and its application to a design by shopping paradigm". In Proceedings of ASME 2003 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2003, Vol. 2003, ASME, pp. 795–804.

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

[16] Deb, K., and Tiwari, S., 2008. "Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization". *European Journal of Operational Research,* ***185***(3), March, pp. 1062–1087.