HAL open science

Multi-objective layout optimization of industrial environment

Xiaoxiao Song, Emilie Poirson, Yannick Ravaut, Fouad Bennis

To cite this version:

Xiaoxiao Song, Emilie Poirson, Yannick Ravaut, Fouad Bennis. Multi-objective layout optimization of industrial environment. 2022 8th International Conference on Optimization and Applications (ICOA), Oct 2022, Genoa, Italy. pp.1-6, 10.1109/ICOA55659.2022.9934713 . hal-03849017

HAL Id: hal-03849017

https://hal.science/hal-03849017v1

Submitted on 11 Nov 2022

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.

Multi-objective layout optimization of industrial environment

Xiaoxiao Song[1], Emilie Poirson[1], Yannick Ravaut[2], Fouad Bennis[1]

***Abstract*— An optimal layout configuration has considerable impacts on industrial engineering. The conventional layout problem aims to find the optimal arrangement of components inside the container while satisfying no component overlap and no container protrusion constraints. In this paper, we address accessibility requirements in layout problems. The novel layout model consists of components with solid and virtual parts. On the one hand, virtual spaces associated with solid components represents the accessibility of component. On the other hand, characterizing accessibility as a constraint ensures components are accessible from the container's entry. The industrial layout involves various objectives and is usually formulated as a multi-objective problem. Hence, a novel multi-objective layout optimization using constructive placement and simulated annealing, named constructive optimization, is developed. A complete layout is generated by constructive placement and evaluated by simulated annealing. It only explores the feasible space and greatly reduces the computational effort. The experimental results prove that the constructive optimization method is effective in solving the problem of industrial layout.**

***Index Terms*— Multi-objective, Industrial layout, Accessibility, Constructive optimization**

I. Introduction

Layout problems (LPs) are inherently multidisciplinary tasks and are usually solved as optimization problems, expressed as finding the optimal arrangements of components inside the container to optimize the objectives and respect constraints. The research on industrial layout optimization intensified in recent years. Industrial layouts have complex environments and various design goals and constraints to ensure that the layout is in a good functional state. For example, different components are placed inside the shelter, as illustrated in Fig. 1. A proper shelter layout can effectively improve the system performance. Light and mobile shelter with on-board equipment provides complete protection for personnel and against battlefield aggression.

In previous researches, the layout model defined by representation of components, formulation of objectives and evaluation of constraints may be different. The component is usually represented by a rectangle or a circle to simplify the problem formulation. Indeed, a majority of studies work on single-objective problem, for example minimizes material handling cost [1] or maximizes adjacency requirement [2]. However, industrial layouts have more than one objective, such as space utilization, activity cost, mass distribution, comfort and so on. There are compromised solutions in the multi-objective problem, widely known as Pareto-front. One can solve the multi-objective as single-objective problem using ε-constraint [3] or weighted-sum method [4], whereas non-dominance based is a more generic way [5]. In addition, geometrical constraints define no component overlap and no container protrusion. More importantly, the accessibility of components is one functional constraint under research [6]. It expresses that the component must be accessible for usability or maintenance.

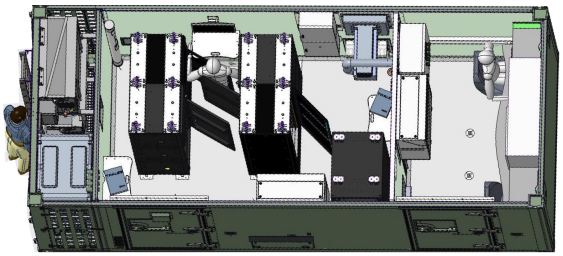


Fig. 1. Shelter layout representation

The discontinuous constraint satisfied region, the non-linear and non-convex objective of layout formulation make the optimization complex in nature. Exact algorithms can optimize small-sized LPs [7]. Stochastic algorithms with global search capabilities are widely used in more complicated industrial applications [8]. For layout optimization, using discrete formulation can reduce the search complexity. The constructive algorithms can generate a complete layout by sequentially placing components [9]. In addition, the constructive algorithm applied with meta-heuristic proved to be an efficient algorithm for layout optimization [10]. For instance, genetic algorithm (GA) coupled with constructive algo-

* This work was supported by China Scholarship Council

[1] X. Song, F.Bennis and E. Poirson are with the Laboratoire des sciences du numérique de Nantes (LS2N), UMR 6004, at École Centrale de Nantes, 44321 Nantes, France.

[2] Y. Ravaut is the Engineer Thales Communications, 49300 Cholet, France.

rithm [11], [12] and simulated annealing (SA) algorithm hybrid with constructive algorithm [13], [14]. SA has fewer parameters and the simpler structure compared to GA optimizations.

The reviewed studies have greatly enriched the layout optimization. Most research work related to constructive optimization was based on the iterative accumulation of objectives to place components. They supposed that the objective function values between components are determined independently. However, these assumptions are not true in reality. Furthermore, accessibility requirements are not completely considered. The main contributions are as follows: It develops a new multi-objective layout formulation with solid and virtual components under accessibility constraints. Because the discontinuous regions mainly cause the difficulty incurred in LP, a constructive optimization is proposed to solve the multi-objective layout model. A layout solution is generated by constructive placement and evaluated by non-domination based SA. Constructive placement circumvents the difficulty arising from constraints and SA conducts the global search ability. The proposed algorithm effectively explores the feasible space and reduces the computational effort.

The rest of the paper is organized as follows. Section II formally defines the layout model. Then constructive optimization framework is described in Section III. Section IV presents the results analysis. Finally, the conclusion and future work are discussed in Section V.

II. Layout problem model

The problem is to locate the rectangular components in a rectangular container. Based on the different functional characteristics of components, the novel component including the solid and virtual part is defined:

- Solid components could not overlap with solid or virtual components and have mass. For example the equipment and device.
- Virtual components could overlap with virtual components and have no mass. For example the accessibility space of desk is virtual and it allows the user sit down, and the space of the cabinet for door opening and closing etc.

Each component is defined by convention $c_i = (s_i, v_{ij})$, s_i is the solid component (solid rectangle), $i = 1, \ldots, n$, n is the number of components; v_{ij} is the associated virtual component (dotted rectangle), $j = 1, \ldots, n_i$, n_i is the number of associated virtual components, as shown in Fig. 2. The solid component s_i is defined by a four-element vector of coordinates and dimensions of the rectangle (x_i, y_i, w_i, h_i). The virtual component v_{ij} is defined by coordinates (x_{ij}, y_{ij}) and size (w_{ij}, h_{ij}) in the local frame of s_i, denoted as $v_{ij} = (x_{ij}, y_{ij}, w_{ij}, h_{ij})$. For

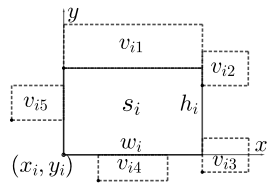


Fig. 2. Component representation

a given number of components, we can evaluate the feasible complexity by estimating the space capacity, which is the most desirable question to inquire in a layout design. The density represents the container area occupied by the components. The density of solid components $\beta_s = \frac{\sum_{i=1}^{n}(w_i \times h_i)}{W \times H}$, and the density of virtual components $\beta_v = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n_i}(w_{ij} \times h_{ij})}{W \times H}$, W, H are the size of the container space. However, the overlap among virtual components is acceptable. A capacity index β_c is defined to measure the minimum occupied space of a given number of solid and virtual components [15]. We can deduce the relationship between the density and capacity as

$$\beta_s \leq \beta_c \leq \beta_s + \beta_v \tag{1}$$

If the LP is feasible, the capacity should be less than 1, the larger the value, the more difficult to find feasible solutions.

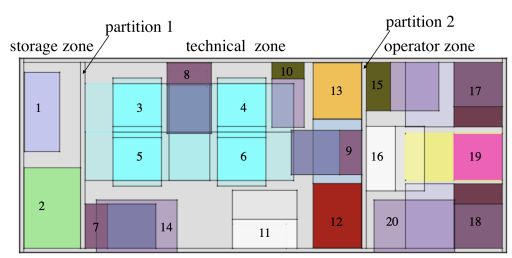


Fig. 3. 2D shelter model

We consider an application to find the optimal layout of facilities in a shelter with three different spaces, named storage zone, technical zone and operator zone, as shown in Fig. 1. For our first study, we only consider the technical zone with a capacity up to 0.82, as the single container optimization problem. We simplify the shelter into two-dimensional because the components are full height, as shown in Fig. 3. Moreover, each component is represented by a rectangle. A set of virtual components attached to components (light color) represent accessibility spaces. For example, virtual spaces of

the cabinet guarantee interaction and correct usability. The model involves two main objective functions: one is to optimize the relative position of components, and the other is to balance the mass distribution, denoted as f_1, f_2 respectively. In the shelter problem, the distance between cabinet 3, cabinet 4, cabinet 5 and energy box 12 should be maximized while the mass distribution should be minimized.

III. Constructive optimization

This section explains the constructive optimization framework summarized in Algorithm 1: a constructive placement algorithm, to place components sequentially with placement sequences $\boldsymbol{c}$ and configuration sequences $\boldsymbol{p}$; a non-domination based SA algorithm, to evaluate the corresponding objective values and optimize sequences $X = (\boldsymbol{c}, \boldsymbol{p})$. The constructive placement circumvents the difficulty arising from constraints and SA conducts the global search. In particular, the constructive optimization is introduced to decrease the computational complexity.

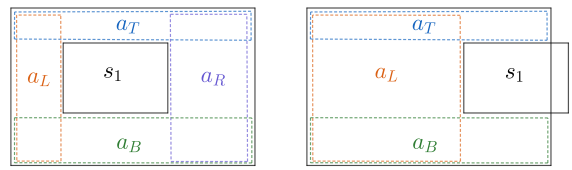
Algorithm 1 Constructive optimization

```
/* Block of SA algorithm */
while stop condition not met do
    Given current state X = (c, p)
    /* Block of constructive placement */
    for c_i in c do
        Select available space by placement strategy
        Place c_i with p_i ∈ p s.t. accessibility analysis
        Space generation
    end
    Objective evaluations F = (f_1, f_2)
    Neighbor generation X_new
end
```

A. Constructive placement

Constructive placement algorithm is inspired from [16]. The former algorithm was developed for the cutting problem and the virtual components are not considered. To place components with respect to constraints, we integrate space generation of solid and virtual components.

1) Space generation: The space around the placed components will be divided into available spaces. The available rectangular space is defined by the vertices of lower left corner, the dimensions along the axes where $a = (x_a, y_a, w_a, h_a)$. The complete space generation between the component space and available space generates four candidate available spaces a_L, a_R, a_T, a_B, as shown in Fig. 4(a). If the component space and available space partially intersect, some candidate available space, for example a_R, may not exist in Fig. 4(b). To place components in the feasible regions, $\boldsymbol{a}$ tracks the available space generation of placed solid components while $\boldsymbol{a}'$ records the available space generation of placed solid and virtual components. In addition, before adding candidate available spaces to the space list, it should remove the available space if it is included in any candidate available space, and filter out the candidate available space if it is included in any available space. The update aims to release storage space. New virtual components

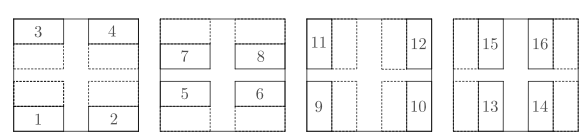


4.(a) Complete included case 4.(b) Partial intersected case

Fig. 4. Space generation

will be placed in $\boldsymbol{a}$ to benefit overlap between virtual components, while new solid components will be placed in $\boldsymbol{a}'$ to guarantee non-overlap of solid components. Placing the new components in available spaces ensures the search for feasible solutions.

2) Placement strategy: For a component c_i, it has four rotation configurations. The placement is performed only for available space in which the component fits. If the selected space $a \in \boldsymbol{a}$ and $a' \in \boldsymbol{a}'$ are coincide, then the component will be placed in the corners of selected space with four rotations. It ensures less margin space is generated. The feasible configurations are numbered from 1 to 16 as shown in Fig. 5, and we have the configuration sequence $p_i = (1, 2, ..., 16)$. Otherwise, the solid component will be placed in the corners of a', where certain configurations will be adjusted according to the selected space a. One example is given in Fig. 6, instead of placing c_2 in the corner in Fig. 6(a), the position is refined to avoid overlapping with c_1 in Fig. 6(b).



5.(a) 5.(b) 5.(c) 5.(d)

Fig. 5. Placement examples a and a' are coincide

The component configuration is decided by the selected space in $\boldsymbol{a}$ and $\boldsymbol{a}'$. The successive placement

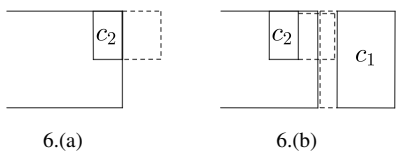


Fig. 6. Placement adjustment

process can be treated as a combination problem. Thus, an effective space selection rule, namely placement strategy, is essential for a constructive placement. In this paper, three placement strategies are proposed and compared in experimental tests.

- Strategy 1: Check all the spaces.
- Strategy 2: Start the placement with the smallest sized space.
- Strategy 3: Try the largest sized space.

3) Accessibility analysis: In the problem modeling, we introduce the virtual components to deal with accessibility. Indeed, the virtual space may be inaccessible if there is no path to access it. Therefore, the proposed method characterizes component accessibility as a constraint during the construction process.

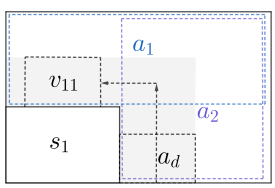


Fig. 7. Connection path [a_d, a_2, a_1]

Assuming one component is accessible from the entrance, there is at least one path for the human to reach the component. The accessibility analysis uses spaces in $\boldsymbol{a} = \{a_1, ..., a_m\}$, m is the number of spaces. When placing a new component into the container, one connection tree is built. The connection is measured by intersection space:

$$\max(0, \min(x_{a_i} + w_{a_i}, x_{a_j} + w_{a_j}) - \max(x_{a_i}, x_{a_j})) \geq w_r \tag{2}$$

$$\max(0, \min(y_{a_i} + h_{a_i}, y_{a_j} + h_{a_j}) - \max(y_{a_i}, y_{a_j})) \geq h_r \tag{3}$$

where $a_i, a_j \in \boldsymbol{a}$, the rectangle size (w_r, h_r) represents the accessible space required by the human. Once the tree generation is finished, check if there is one path for accessibility. The path starts from the entrance space a_d and ends at the virtual space of the component. One example presented in Fig. 7, the connection tree is $\{a_d : [a_2], a_2 : [a_1], a_1 : []\}$, the placement of v_{11} occupies a_1, and there is one path=[a_d, a_2, a_1].

The placement of component c_i may have more than one feasible configurations satisfying the constraints. We need the criteria to select which configuration is used for the space generation. To enhance the feasibility, we classify the configurations based on the container boundary then select configuration according to the overlap maximization rule.

B. SA algorithm

With sequences $X = (\boldsymbol{c}, \boldsymbol{p})$, the placement algorithm can constructively build a layout. Thus, the proposed algorithm is based on the combinatorial problem. SA is a stochastic neighborhood search approach for global optimization and has been successfully applied to various combinatorial problems. In previous work, we proposed a non-domination based SA [15] to solve multi-objective problems and proved its good performance. So it is used to improve the sequences X here. An external archive is used to keep non-dominated solutions during the optimization. And the neighbors are generated by a swap operator. In the swap procedure of the placement sequence, σ is related to temperature t:

$$\sigma = n * exp^{(-1/t)} \tag{4}$$

The integer parameter $\sigma \in [1, n]$ determines the process of neighbor generation. With high temperature and big σ, any two elements of the sequence can be exchanged; with low temperature and small σ, only the last few elements could be exchanged. The mechanism is the same in the configuration sequence. Given a state X, a layout with $F(X)$ is generated using the constructive placement. In SA, we consider the new state X_{new} as a better solution based on the non-domination relationship between $F(X_{new})$ and $F(X)$. Assuming that all objective functions are minimized, $F(X_{new})$ dominates $F(X)$ if $F(X_{new})$ is no worse than $F(X)$ for all objectives and $F(X_{new})$ is better than $F(X)$ for at least one objective. Then definitely X_{new} is a better solution. However, accepting a poor solution enables uphill moves sometimes. The new state X_{new} will replace the current state X if one of the conditions is satisfied:

1) $F(X_{new})$ dominates $F(X)$.
2) $F(X)$ dominates $F(X_{new})$, $rand(0,1) < exp^{(-(F(X_{new})-F(X))/t)}$
3) $F(X_{new})$ and $F(X)$ are non-dominated solutions. $F(X_{new})$ is not dominated by any solution in the archive; or $F(X_{new})$ is not far from the obtained Pareto-front.

The optimization searches for new solutions until it reaches the maximum iterations L.

IV. Experiment result

In this section, three different examples are formulated to assess the developed constructive placement. Then the constructive optimization is applied to solve the practical shelter problem.

A. Test examples

Here, we design three layout examples to test the different placement strategies. It is worth noting that there is no objective but to verify the exploration ability under constraints. In other words, with a given number of iteration, the more feasible solutions it find, the better the performance. The properties of the example are summarized in TableI. Indeed, all the test examples have high feasible complexity due to the dense capacity.

TABLE I

Properties of test examples

Property	Test 1	Test 2	Test 3
Number of components	18	11	9
Density of solid components	0.54	0.47	0.38
Density of virtual components	0.54	0.66	0.65
Density of solid, virtual components	1.08	1.13	1.03
Capacity	0.81	0.75	0.77
Equal size	Yes	No	No
Accessibility	No	Yes	Yes
Number of components edge on the wall	0	0	2

1) Test 1 – Equal-sized component: The equal-sized component can eliminate the effect of different placement sequences. There are only geometrical constraints.

2) Test 2 – Unequal-sized component: The unequal-sized component is more common and realistic. Consider the accessibility requirement and the geometrical constraints.

3) Test 3 – Big-sized component: The quite big-sized component introduces the size difference among the unequal-sized components. Integrate the additional constraints such as edge on the wall, accessibility constraints and the geometrical constraints.

The strategy comparison results are summarized in TableII where the number of solutions are obtained with a given number of iterations. Three strategies are compared with fixed/permuted configuration sequence. By comparison, we can conclude that

- The permutation of configuration sequence is necessary for the diversity in the design space, especially in the case of equal-sized component.
- Considering the search ability under constraints, strategy 2 is much better than strategy 3. The edge constraints improve the performance of strategy 1 but it is time consuming. Strategy 2 conducts the placement effectively and generates a number of feasible solutions.

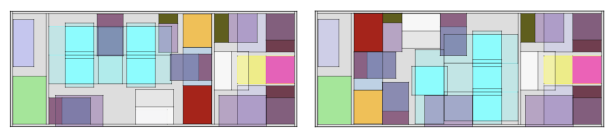
TABLE II

Strategy comparison

Number of solutions		Test 1	Test 2	Test 3
Fixed p_i	Strategy 1	1/100	2/100	24/600
	Strategy 2	1/100	37/100	21/600
	Strategy 3	1/100	5/100	2/600
Permuted p_i	Strategy 1	22/100	5/100	51/600
	Strategy 2	26/100	40/100	50/600
	Strategy 3	13/100	6/100	18/600

B. Practical example

Based on the above analysis, strategy 2 with configuration permutation proved to be the best placement strategy for developing the constructive optimization algorithm.



8.(a) Initial solution 8.(b) Optimal solution

Fig. 8. Solution in design space

TABLE III

Optimal solution and initial solution

Objective	Initial solution	Optimal solution
f_1 (maximization)	7164.6	8003.9
f_2 (minimization)	74.3	47.4

Through limited iterations $L = 4000$, the algorithm generates 337 configurations for the shelter. The algorithm searches for solutions by considering the geometrical and functional constraints and objectives of the problem formulation. Among these solutions, the expert may choose, for instance, the solution realizes the best compromise between optimization objectives. Fig. 8 shows the initial solution and an optimal solution. It can be seen that there is a significant difference between the optimal layout solution and the initial solution. Table III illustrates that the optimal solution can realize much better objective values compared to the initial one. The experimental results prove that the proposed optimization algorithm is effective in solving the LP in the industry.

V. Conclusion

In most industrial layout applications, the discontinuous constraint satisfied region, the non-linear and non-convex objective of layout formulation make the optimization complex in nature. This paper deals with the LP with solid and virtual components under accessibility constraints. It increases the computation complexity, and needs an effective method to optimize the new layout model. A non-domination based constructive optimization method is proposed to solve the novel multi-objective layout model. Layout solutions are generated by constructive placement and evaluated by non-domination based SA algorithm. The strategy comparisons confirm that placement can effectively generate feasible solutions according to placement and configuration sequences. The SA search technique explores the layout configurations and conducts the global search ability. The experimentation indicates the efficiency of constructive optimization in finding high-qualified industrial layout solutions.

The proposed algorithm assumes having one container space. Further research could extend the algorithm to multi-container LPs. It could be interesting if the space division (the partition in the shelter problem) is formulated as one variable to ensure the automatic layout design.

Acknowledgment

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

References

[1] Z. Lin and Z. Yingjie, "Solving the facility layout problem with genetic algorithm," in *Proceedings of IEEE International Conference on Industrial Engineering and Applications (ICIEA)*, 2019, pp. 164–168.

[2] F. Ghassemi Tari and H. Neghabi, "A new linear adjacency approach for facility layout problem with unequal area departments," *Journal of Manufacturing Systems*, vol. 37, pp. 93–103, 2015.

[3] A. Che, Y. Zhang, and J. Feng, "Bi-objective optimization for multi-floor facility layout problem with fixed inner configuration and room adjacency constraints," *Computers & Industrial Engineering*, vol. 105, pp. 265–276, 2017.

[4] H. Zhang, A. ping Li, and X. mei Liu, "Modeling and optimization for hybrid layout design with behavioural constraints," in *Proceedings of International Conference on Mechatronics and Automation*, 2009, pp. 5052–5056.

[5] A. Kaveh, M. Rastegar Moghaddam, and M. Khanzadi, "Efficient multi-objective optimization algorithms for construction site layout problem," *Scientia Iranica*, vol. 25, no. 4, pp. 2051–2062, 2018.

[6] X. Song, E. Poirson, Y. Ravaut, and F. Bennis, "Interactive design optimization of layout problems," in *Proceedings of IFIP International Conference on Advances in Production Management Systems*, 2021, pp. 387–395.

[7] E. Ouassam, N. Hmina, B. Bouikhalene, and H. Hachimi, "Heuristic methods: Application to complex systems," in *Proceedings of the 7th International Conference on Optimization and Applications*, 2021, pp. 1–8.

[8] Q. Zheng and Y.-G. Liang, "A study on the particle swarm optimization with adaptive weight constrained layout optimization," in *Proceedings of the 8th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2016, pp. 283–287.

[9] H. Hosseini nasab, S. Fereidouni, S. Ghomi, and M. Fakhrzad, "Classification of facility layout problems: a review study," *The International Journal of Advanced Manufacturing Technology*, vol. 94, pp. 957–977, 2018.

[10] F. Halawa, S. Chalil Madathil, and M. T. Khasawneh, "Integrated framework of process mining and simulation–optimization for pod structured clinical layout design," *Expert Systems with Applications*, vol. 185, no. 12, pp. 1–17, 2021.

[11] J. F. Gonçalves and M. G. Resende, "A biased random-key genetic algorithm for the unequal area facility layout problem," *European Journal of Operational Research*, vol. 246, no. 1, pp. 86–107, 2015.

[12] J. Huo, J. Liu, and H. Gao, "An nsga-ii algorithm with adaptive local search for a new double-row model solution to a multi-floor hospital facility layout problem," *Applied Sciences*, vol. 11, no. 4, pp. 1758–2753, 2021.

[13] Y. Xiao, Y. Seo, and M. Seo, "A two-step heuristic algorithm for layout design of unequal-sized facilities with input/output points," *International Journal of Production Research*, vol. 51, no. 14, pp. 4200–4222, 2013.

[14] M. Z. Allahyari and A. Azab, "Mathematical modeling and multi-start search simulated annealing for unequal-area facility layout problem," *Expert Systems with Applications*, vol. 91, pp. 46–62, 2018.

[15] X. Song, E. Poirson, Y. Ravaut, and F. Bennis, "Efficient multi-objective simulated annealing algorithm for interactive layout problems," *International Journal on Interactive Design and Manufacturing*, vol. 15, no. 4, pp. 441–451, 2021.

[16] K. Lai and J. W. Chan, "Developing a simulated annealing algorithm for the cutting stock problem," *Computers & Industrial Engineering*, vol. 32, no. 1, pp. 115–127, 1997.