A Modified Simulated Annealing and Enhanced Harmony Search Algorithms for the Continuous Facility Layout Problem (CFLP)

SOUMAYA LAKEHAL[1], ABDELHAKIM AITZAI[2]

[1]National Higher School of Artificial Intelligence ENSIA
16201 Algiers,
ALGERIA

[2]RECITS Laboratory, USTHB
16042 Algiers,
ALGERIA

Abstract: - This paper focuses on solving the Continuous Facility Layout Problem (CFLP), which aims to minimize material handling costs by strategically placing facilities in a known or unknown area. The objective is to find optimal or near-optimal facility arrangements while adhering to non-overlapping and spatial constraints, thus enhancing the efficiency of production systems. The study addresses various layout scenarios, including single-row facility layout (with and without clearance), continuous layout, and unequal-area facility layout problems. To achieve this, the authors propose a mixed-integer nonlinear programming (MINLP) model tailored for continuous layouts. Two meta-heuristic algorithms are developed to optimize this model: a hybrid Simulated Annealing-Genetic Algorithm (SA-GA), which leverages genetic crossover operations, and an Enhanced Harmony Search Algorithm (EHSA), featuring dynamic parameters and a novel improvisation technique. The proposed methods provide flexible and efficient solutions for both small and large-scale layout problems, offering decision-makers practical tools for real-world applications.

Key-Words: - Continuous Facility Layout Problem, Simulated Annealing, Harmony Search Algorithm, Meta-heuristics, Optimization, Decision algorithms, VIP-PLANOPT.

Received: May 2, 2024. Revised: September 18, 2024. Accepted: October 19, 2024. Published: November 14, 2024.

1 Introduction

Efficient organization and facility layout can significantly reduce operational costs related to product and material handling. Solving Facility Layout Problems (FLPs) involves determining the most effective arrangement of a set of facilities—whether machines, load centers, or departments—on the floor of a productive system. These arrangements must satisfy predefined objectives set by decision-makers, while also adhering to constraints, such as spatial limitations and non-overlapping requirements. In the Continuous Facility Layout Problem (CFLP), the facilities have unequal areas, are placed anywhere on the floor, and must not overlap or exceed the boundaries of the placement area.

Several types of facility layout problems exist, varying based on the characteristics of the production system. Among these, decision-makers often encounter four key configurations: the single-row facility layout problem, where facilities are aligned along a single line; the multi-row facility layout problem, where facilities are arranged across multiple rows; the fixed-location facility layout problem (QAP), where facilities are assigned to pre-defined locations; and the open-field layout problem, known as the continuous layout problem (CLP), in which facilities are free to be placed anywhere in a continuous space.

The CFLP is computationally intensive, especially when the number of facilities is large or when their dimensions vary significantly. Many researchers have proposed heuristic and meta-heuristic approaches to solve continuous layout problems; however, these methods often struggle to provide efficient solutions for large-scale problems. Consequently, there remains a need for methods that strike a balance between computational efficiency and solution quality. This paper proposes two meta-heuristic approaches—a hybrid Simulated Annealing-Genetic Algorithm (SA-GA) and an Enhanced Harmony Search Algorithm (EHSA)—to address this issue. These methods aim to improve the quality of solutions while reducing the computational time required for large-scale problems.

The structure of this paper is as follows: Section 2 presents the mathematical formulation of the CFLP, explaining the objectives and constraints in detail. Section 3 introduces the proposed meta-heuristic algorithms (SA-GA and EHSA), including their design and implementation. Section 4 provides a comprehensive evaluation of the algorithms, presenting experimental results based on benchmark datasets. Section 5 discusses the implications of the results and highlights why certain algorithms perform better on specific problem instances. Finally, Section 6 concludes the paper by summarizing the contributions and suggesting areas for future research.

2 Litterature Review

Each mathematical formulation of the layout problem is based on certain hypotheses, some formulations consider an arrangement on fixed sites, others consider an arrangement on well-defined lines, blocks or areas, and some consider a space layout.

According to the geometric characteristics of the studied problem, *FLPs* can be classified into numerous types, the most known bases of classification are: *Equal* Vs *Unequal* areas facilities, and *Open* Vs *Closed* placement filed. In the closed field FLPs, single-row FLP, [1], [2], and multi-rows FLP, [3], [4], are the major cases. While in the open field FLPs, [5], there is no constraints on the enclosing area.

Articles in the literature refer to two categories of modeling, the *discrete QAP* (Quadratic Assignment Problems) and the *MIP* (Mixed-Integer Programming) models. For discrete QAP modeling, [3], *N* facilities and *N* locations are considered. For each pair of locations, a distance between them is specified and for each installation, the flow through it is determined. The problem then is to assign an installation for each location, minimizing the sum of material handling between two locations and the corresponding flows. For MIP models, [5], the objective function is of the same nature as in the previous case, but is based on the X and Y coordinates of facilities centroids, the distance depends on these coordinates and it is calculated by using Euclidean or Rectilinear distance. Note that it can be very difficult to linearize certain constraints and the same thing in the discrete case in QAP.

2.1 Unequal-Area Facility Layout Problem (UA-FLP)

Most studies on the facility layout problem suppose a common hypothesis: all departments are of equal size. Other studies do not consider this Hypothesis. Examples of such studies are [6], [7], [8].

In unequal area FLP studies, the representation of the layout is continuous; this type of arrangement is often formulated by mixed integer programming, [9]. In the continuous layout all machines are placed anywhere on the site and should not overlap, the placement variables of a machine i are either the coordinates of its centroid (x_i, y_i) , such that its half-length $\frac{l_i}{2}$ and its half-width $\frac{w_i}{2}$ are known, [10], or by the coordinates from the bottom left (x_i^l, y_i^l) , such that its length l_i and its width w_i are known. A study of the placement of rectangular facilities in an unlimited floor space without overlap is presented in [11].

The unequal-Area facility layout problem is NP-Hard because of the complexity of its solution.

2.2 Resolution Approaches for the Unequal-Area Faclity Layout Problem (UA-FLP)

Unequal-area facility layout problems have been solved using various meta-heuristics methods that have been successfully applied on the problem.

The study in [12], reported an effective memetic search algorithm to solve UA-FLP, a development of a hybrid genetic algorithm is reported in [11]. The research introduced in [13], presented a modified genetic search based on a local search algorithm for solving static FLP with unequal compartments. Recently, a novel island model for solving UA-FLP was proposed in [14]. In Simulated annealing algorithm was efficiently applied on the UA-FLP, in [15], it was proposed simulated annealing algorithm for placing manufacturing cells considering areas and shapes requirements. [16], reports an application of simulated annealing for solving UA-FLP with a flexible bay structure. In [17], a memory-based simulated annealing algorithm called the Dual Memory Simulated Annealing Algorithm (DMSA) is presented to solve multi-line facility layout problems. [18], proposes a novel heuristic approach, sequential solution method (SSM), for the efficient solution of Continuous Facility Layout Problems.

From existing papers in the literature, we can observe that few studies used a modified simulated annealing and an enhanced harmony search algorithm for solving continuous facility layout problem. Simulated annealing is a successful and widely used method for combinatorial optimization. However, it could be trapped in a local minimum, the classical simulated annealing lacks exploration. To deal with the disadvantage of premature convergence the neighborhood search is enhanced with the crossover technique that explores the search space efficiently.

In the harmony search algorithm, we consider a dynamic setting of parameters to save good harmonies in the memory during the search procedure. The random nature of improvisation may causes the generation of worse harmonies, so there will be no improvement in the harmony memory, and then the algorithm converges slowly to good solutions. To escape this drawback, we integrate the crossover operator in improvisation to permit best harmonies to transmit their parts, and in the same time, to avoid falling into local minimum and the stagnation of the method. The proposed methods SA-GA and EHSA provide very good results.

Contributions of the paper

The contributions of the paper are:

- A new study that can be applied on facility layout problems with all characteristics of the placing area.
- A new method SA-GA that integrates the crossover operator of genetic algorithm in simulated annealing process.
- A new technique of improvisation in harmony search algorithm based on the crossover operator.

3 Problem Definition and Formulation

The problem consists of placing N facilities of different sizes, where the dimensions of each facility are known. The facilities are placed within a space of length L and width W. If the exact placement dimensions are not specified, a virtual space dimension is defined, as shown in Equation (1). Each facility i is assigned horizontal (x_i) and vertical (y_i) coordinates, corresponding to the top-left point of the facility on the X and Y axes (Fig. 1).

$$L = \text{random}\left(\frac{\sum_{i=1}^{N} l_i}{2}, \frac{\sum_{i=1}^{N} l_i}{3}\right) \tag{1}$$

Equation (1) defines the length of the placement area as a random value within the given range, based on the total length of all facilities. This random length helps define the available floor space if specific dimensions are not provided.

Facilities are placed while satisfying several constraints: non-overlapping of facilities, ensuring safety distances between them and ensuring that no facility exceeds the boundaries of the placement space.

The objective is to minimize the material handling cost, calculated by multiplying the flow of materials between facilities by the distance between them.

Notations used in the problem:

Indices:

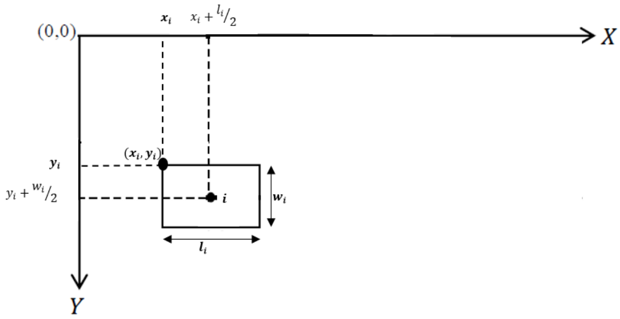

Fig. 1: Facility placement in the floor space

- i, j: Facility indices.

Data:

- N: The number of facilities.
- L: The length of the placement space.
- W: The width of the placement space.
- ds: Minimum clearance distance between facilities.
- l: A vector containing the lengths of all facilities.
- w: A vector containing the widths of all facilities.
- l_i: The length of facility i.
- w_i: The width of facility i.
- C_{ij}: Flow cost between facility i and facility j.

Variables:

- x_i: X-coordinate of facility i.
- y_i: Y-coordinate of facility i.
- D_{ij}: Distance between facility i and facility j.

The objective function of the problem is:

$$\text{Minimize } Z = \sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij} D_{ij} \tag{2}$$

Equation (2) represents the total material handling cost, which is calculated as the sum of flow costs C_{ij} between facilities i and j, multiplied by the distance D_{ij} between their centroids.

Constraints:

Non-overlapping constraints for the X-axis:

$$\left|\left(x_i + \frac{l_i}{2}\right) - \left(x_j + \frac{l_j}{2}\right)\right| \geq \frac{l_i}{2} + \frac{l_j}{2} + ds \tag{3}$$

Equation (3) ensures that the facilities do not overlap horizontally. It checks that the distance between the horizontal centroids of facilities i and j is greater than or equal to the sum of their half-lengths plus the minimum clearance distance ds.

Non-overlapping constraints for the Y-axis:

$$\left|\left(y_i + \frac{w_i}{2}\right) - \left(y_j + \frac{w_j}{2}\right)\right| \geq \frac{w_i}{2} + \frac{w_j}{2} + ds \quad (4)$$

Equation (4) ensures that the facilities do not overlap vertically. Similar to Equation (3), it checks that the distance between the vertical centroids of facilities i and j is sufficient to prevent overlap, accounting for the half-widths of the facilities and the clearance distance ds.

Boundary constraints for the X and Y dimensions:

$$x_i + l_i \leq L \quad (5)$$

$$y_i + w_i \leq W \quad (6)$$

Equations (5) and (6) ensure each facility stays within the placement space. They guarantee that the right edge ($x_i + l_i$) and the bottom edge ($y_i + w_i$) of each facility do not exceed the boundaries of the available length L and width W.

Distance calculation:

$$D_{ij} = \left|\left(x_i + \frac{l_i}{2}\right) - \left(x_j + \frac{l_j}{2}\right)\right| + \left|\left(y_i + \frac{w_i}{2}\right) - \left(y_j + \frac{w_j}{2}\right)\right| \quad (7)$$

Equation (7) calculates the distance D_{ij} between the centroids of facilities i and j using the rectilinear (Figure 2) distance formula, which sums the absolute differences in their X and Y coordinates.

Non-negative constraints for facility coordinates:

$$x_i, y_i \geq 0 \quad (8)$$

Equation (8) ensures that each facility's X and Y coordinates are non-negative, meaning all facilities are placed within the valid area of the placement space.

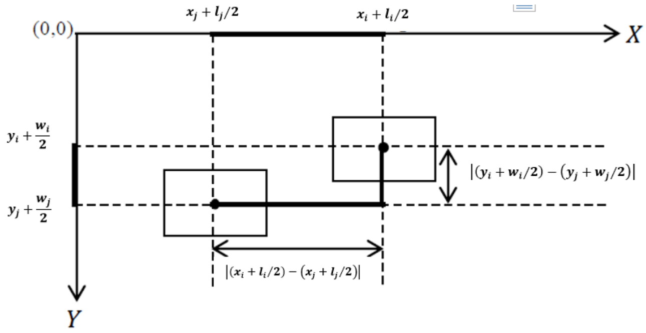

Fig. 2: Illustration of the rectilinear distance between facilities

4 Resolution Approaches

Because of the high complexity of the problem due to the combinatorial explosion of the solution space, the proposed resolution method is approximated and based on meta-heuristics.

Two hybrid meta-heuristics are proposed for solving the studied problem. The first method is a local search meta-heuristic based on the hybridization of Simulated Annealing (SA) and the crossover operation of the Genetic Algorithm (GA), referred to as SA-GA. The second meta-heuristic is an Enhanced Harmony Search Algorithm (EHSA) with dynamic parameters, where the improvisation operation is applied based on the crossover principle of the Genetic Algorithm.

4.1 Solution Encoding

In this work, we have chosen to implement a direct encoding of solutions. A solution is coded as a sequence (vector) of N elements, where N is the number of facilities. The position of a facility in the sequence determines the order in which it is assigned to the next feasible coordinates (Fig. 3).

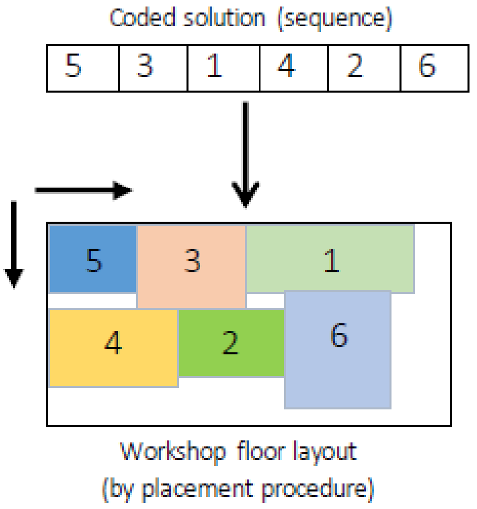

Fig. 3: Solution coding

4.2 The Initial Solution

The initial solution is represented by a feasible sequence of facilities generated randomly. The configuration of the generated solution is determined by placing the next facility in the sequence at the next feasible coordinates (Fig. 4).

4.3 Hybrid Simulated Annealing-Genetic Algorithm (SA-GA)

The hybridization of Simulated Annealing (SA) with the Genetic Algorithm (GA) enhances the

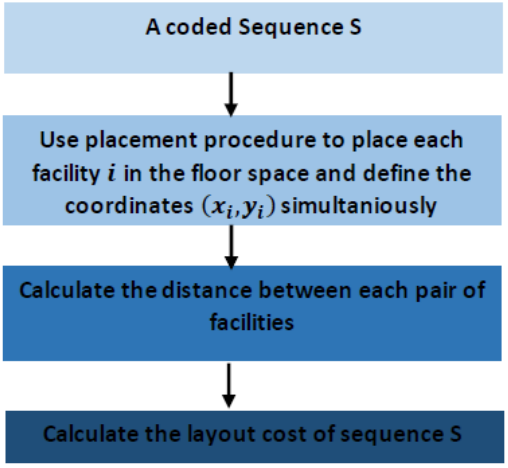
A coded Sequence S
Use placement procedure to place each facility i in the floor space and define the coordinates (x_i, y_i) simultaniously
Calculate the distance between each pair of facilities
Calculate the layout cost of sequence S

Fig. 4: Solution sequence decoding procedure

neighborhood structure in the search process and contributes to improving the exploration of the solution space.

SA-GA integrates a genetic algorithm into the Simulated Annealing search process. The SA generates an initial solution that satisfies the problem constraints, initializes the temperature, and starts the search. At each iteration, the current best solution is evaluated, and a population of P neighboring solutions is generated using SA's neighborhood operators. The GA then applies its genetic operators (such as crossover and mutation) on this population to produce a new generation. After each generation, the best individual from the GA population is returned to SA for further improvement. This process continues until the termination condition is reached, which is usually the final temperature T_f.

4.4 Enhanced Harmony Search Algorithm (EHSA)

The Harmony Search Algorithm (HSA) is a meta-heuristic developed by [19], and it is confirmed that it is inspired by the process of musical performance. In musical performance, musicians continuously improve harmonies by playing notes, adjusting each note to achieve the perfect harmony. Similarly, in HSA, each decision variable in the optimization process is treated as a musician playing a note, and the goal is to find the best combination of these variables (harmonies) that leads to the optimal solution.

HSA has been successfully applied to several optimization problems such as the Traveling Salesman Problem (TSP), [20], and the Vehicle Routing Problem (VRP).

In the Enhanced Harmony Search Algorithm (EHSA), a new solution generation technique is applied during the improvisation step. While the algorithm progresses, if a certain probability condition is met, the new solution is generated from harmony memory. A specific strategy is applied for value selection in this memory.

The harmony memory is sorted according to the objective function's increasing order, and it is then divided into two sub-memories of size HMS/2. The first sub-memory, HM_{best}, contains the top HMS/2 best harmonies, while the second sub-memory, HM_{worst}, contains the remaining worst harmonies. The values of the first $N/2$ notes $(s_i^{new}, i = 1, \ldots, N/2)$ in the new harmony are selected from HM_{best}, while the last $N/2$ notes $(s_i^{new}, i = (N/2) + 1, \ldots, N)$ are chosen from HM_{worst}.

The harmony memory is updated to obtain a feasible solution, and the objective function is recalculated for the new solution.

5 Expiremental Results

5.1 Data Set Description

The performance of the proposed algorithms is evaluated using three sets of problems selected from the literature, consisting of 33 problems in total. The first set of problems includes 13 single-row facility layout problem (SRFLP) instances. The second set, called SR-CL, includes eight single-row facility layout problems that account for clearance between facilities. The third set, called UAF, includes 12 unequal-area facility layout problems (UA-FLP). The details of the data sets are summarized in Table 1 in Appendix A.

5.2 Results Comparison and Analysis

We compared the performance of the proposed SA-GA and EHSA algorithms on the datasets, evaluating them by the average best-found solution and the average deviation from the best-known solution. The SA-GA algorithm was run 200 times independently, while the EHSA algorithm was executed 200 times for various iterations: $50, 100, 150, 200, \ldots, 500$.

Explanation of Table 2: Table 2 in Appendix A presents the computational results of the proposed methods in comparison to the standalone Simulated Annealing (SA) and Genetic Algorithm (GA). The best-known solutions from the literature are also provided. The SA-GA and EHSA results are evaluated against these known solutions. Additionally, the table includes the average CPU execution time in seconds over 200 runs of each algorithm.

Key Insights from Table 2:

For small and medium-sized instances (e.g., $N5-01$, $N6-01$), SA-GA achieves optimal solutions quickly, as evidenced by its lower execution time compared to EHSA. However, EHSA consistently provides better or equally optimal solutions, especially for larger and more complex instances (e.g., $Sko64-01$, $Sko100-01$). EHSA outperforms SA-GA in terms of solution quality for larger instances, where the dynamic parameters and enhanced improvisation of EHSA allow it to avoid premature convergence. However, EHSA requires more computation time, especially on large-scale problems.

Expanded Discussion:

SA-GA performs well for small instances due to its fast convergence, which is driven by the genetic algorithm's crossover operations and the efficient exploration of the search space. This allows SA-GA to find good solutions quickly without requiring extensive computational time.

For larger instances, EHSA excels in solution quality due to its dynamic parameter setting and novel improvisation strategy, which provide more robust exploration and prevent the algorithm from getting stuck in local optima. This explains why EHSA outperforms SA-GA on larger datasets like $Sko100-01$, despite taking longer computation times.

Table 3 in Appendix A presents the computational results for the SR-CL dataset, which includes instances of the Single-Row Facility Layout Problem with clearance considerations. In this table, we compare the performance of the proposed algorithms—SA-GA and EHSA—against the best-known solutions for each problem instance. We also provide the execution time for each algorithm in seconds (t(s)), which allows us to evaluate the trade-off between solution quality and computational efficiency.

Key Observations from Table 3

Solution Quality:

For all the problem instances in the SR-CL dataset, both SA-GA and EHSA achieve the best-known solution (denoted as S#). This demonstrates that both algorithms are effective at solving these types of layout problems and can consistently produce optimal solutions.

Execution Time:

While both algorithms achieve the same solution quality, SA-GA significantly outperforms EHSA in terms of execution time. For example, in the $C20$ instance, SA-GA finds the best solution in 7.34 seconds, whereas EHSA takes 11.34 seconds. This pattern is consistent across all instances, with SA-GA generally taking less time than EHSA.

Interpretation:

The results in Table 3 (Appendix A) indicate that while both algorithms are capable of solving the SR-CL problems to optimality, SA-GA is more computationally efficient. This is primarily because SA-GA relies on genetic algorithm crossover operations that accelerate the search process, particularly for smaller and medium-sized problems. In contrast, EHSA is a more exploration-focused algorithm, which explains why it takes longer to converge to the same solution. EHSA uses dynamic parameter settings and an enhanced improvisation process to explore the search space more thoroughly. While this leads to better results for larger and more complex instances (as seen in Table 4 in Appendix A), it can result in longer computation times for smaller datasets like SR-CL.

Expanded Comments:

SA-GA is more suitable for scenarios where computational speed is critical, such as real-time decision-making systems. EHSA, although slower, provides more robust performance in larger and more complex datasets, as it avoids local minima and ensures a thorough search of the solution space. Thus, for the SR-CL dataset, SA-GA would be preferred if execution time is a priority, while EHSA might be more appropriate for larger-scale problems requiring deeper exploration, which we discuss further in Table 4 (Appendix A).

To study the performance of SA-GA and EHSA algorithms on the dataset UAF, we compare it to two of the best-performing algorithms in the literature that can be found in [24], [26].

In Tab. 4, the best costs S_v and SMI found by [24], [26], respectively are reported.

We report in Tab. 4 in Appendix A, the **average best-found solution** by SA-GA and EHSA.

The results in Tab. 4 show that EHSA outperforms SA-GA in terms of best-found solution except for instance L-008-IM and L010. For instance L003 and L004, the best-found solutions by both SA-GA and EHSA are optimal and equal to the ones of [24], [26]. The results also prove that for instances L008-IM, L010, and L062, the values of the solutions obtained by SA-GA and EHSA outperform the best layout cost previously known for [24], [26]. For instances L050 and L006, the performance of SA-GA and EHSA compared to [24], is also better in terms of solution quality, a cost of **3379** found by VIP-PLANOPT while the layout cost found by both SA-GA and EHSA is **3351.5** for L006 instance. In addition, the best layout cost reported in VIP-PLANOPT is **78224.7**, while the best-found solutions by SA-GA and EHSA are **77943.123** and **77897.768** respectively. For instances L020-MI and L028, the value of the best layout cost found by Mir and Imam is improved from **1199.5** to **1199.049** and **1198.5** by SA-GA and EHSA respectively, while for L020 instance the value of the best layout cost found by [26], is improved from **7536.8** to **7331.5** and **7302.812** by SA-GA and EHSA respectively. For large instances of 100 and 125 facilities (L100 and L125-B), the solution quality is quite well and still good and near to the solutions found by [24], [26].

6 Conclusion

We presented in this paper the resolution of the continuous facility layout problem with equal/unequal area facilities, this research considers the minimization of material handling cost.

We developed two meta-heuristics for solving the continuous facility layout problem. The crossover operator of the genetic algorithm is used as a neighborhood search in the proposed simulated annealing algorithm, and it is implemented in the improvisation technique for the harmony search algorithm. The parameters of the presented harmony search algorithm are set dynamically.

The results of our methods are compared to different data sets from the literature that comprise single-row facility layout problem instances and unequal area facility layout problem instances. The experimental results demonstrate that the suggested resolution approaches are able to provide optimal layouts for the SRFLP in a reasonable computational time, while the layouts obtained for the UA-FLP are very good solutions with low deviations from the best-known solution.

The application of exact methods to solve the continuous facility problem is a very promising and interesting research area, it can be a subject for further studies and experimentation in the future.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT in order to improve readability and language. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References:

[1] D.M. Simmons, “One-dimensional space allocation: an ordering algorithm,” Operations Research, vol. 5, 1969, pp. 812–826.

[2] S. Liu, Z. Zhang, C. Guan, L. Zhu, M. Zhang and P. Guo, “An improved fireworks algorithm for the constrained single-row facility layout problem,” International Journal of Production Research, vol. 8, 2021, pp. 2309–2327.

[3] S.S. Heragu and A. Kusiak, “Efficient models for the facility layout problem,” European Journal of Operational Research, vol. 1, 1991, pp. 1–13.

[4] X. Wan, X. Zuo, X. Li and X. Zhao, “A hybrid multiobjective grasp for a multi-row facility layout problem with extra clearances,” International Journal of Production Research, 2020, pp. 1–20.

[5] S. Das, “A facility layout method for flexible manufacturing systems,” The International Journal of Production Research, vol. 2, 1993, pp. 279–297.

[6] M.J. Wang, M.H. Hu and M.Y. Ku, “A solution to the unequal area facilities layout problem by genetic algorithm,” Computers in Industry, vol. 2, 2005, pp. 207–220.

[7] S. Ingole and D. Singh, “Unequal-area, fixed-shape facility layout problems using the firefly algorithm,” Engineering Optimization, vol. 7, 2017, pp. 1097–1115.

[8] M. Pires, E. Silva and P. Amorim, “Solving the grocery backroom layout problem,” International Journal of Production Research, vol. 3, 2021, pp. 772–797.

[9] S. Kulturel-Konak, A. Konak, “Linear programming based genetic algorithm for the unequal area facility layout problem,” International Journal of Production Research, vol.14, 2013, pp 4302-–4324.

[10] R.D. Meller, V. Narayanan, P.H. Vance, “Optimal facility layout design,” Operations Research Letters, vol.3-5, 1998, pp 117-–127.

[11] F.G. Paes, A.A. Pessoa and T. Vidal, “A hybrid genetic algorithm with decomposition phases for the unequal

area facility layout problem," European Journal of Operational Research, vol. 3, 2017, pp. 742–756.

[12] R.D. Meller, V. Narayanan and P.H. Vance, "Optimal facility layout design," Operations Research Letters, vol. 3-5, 1998, pp. 117–127.

[13] R.K. Hasda, R.K. Bhattacharjya and F. Bennis, "Modified genetic algorithms for solving facility layout problems," International Journal on Interactive Design and Manufacturing (IJIDeM), vol. 3, 2017, pp. 713–725.

[14] L. Garcia-Hernandez, L. Salas-Morera, C. Carmona-Munoz, J. Garcia-Hernandez and S. Salcedo-Sanz, "A novel island model based on coral reefs optimization algorithm for solving the unequal area facility layout problem," Engineering Applications of Artificial Intelligence, 2020.

[15] K.Y. Tam, "A simulated annealing algorithm for allocating space to manufacturing cells," The International Journal of Production Research, vol. 1, 1992, pp. 63–87.

[16] I.B. Hunagund, V.M. Pillai and U. Kempaiah, "Solving unequal area facility layout problems with flexible bay structure by simulated annealing algorithm," In: Advances in Production and Industrial Engineering, 2021, pp. 75–101.

[17] K. Zolfi, J. Jouzdani and H. Shirouyehzad, "A novel memory-based simulated annealing algorithm to solve multi-line facility layout problem," Decision Science Letters, vol. 12, 2023, pp. 69–88.

[18] M.B. Şenol and E.A. Murat, "A sequential solution heuristic for continuous facility layout problems," Annals of Operations Research, vol. 320, 2023, pp. 355–377.

[19] Z.W. Geem, J.H. Kim and G.V. Loganathan, "A new heuristic optimization algorithm: harmony search," Simulation, vol. 2, 2001, pp. 60–68.

[20] M.F. Anjos and A. Vannelli, "Computing globally optimal solutions for single-row layout problems using semidefinite programming and cutting planes," INFORMS Journal on Computing, vol. 4, 2008, pp. 611–617.

[21] R. Love, and J. Wong, "On solving a onedimensional space allocation problem with integer programming, " INFOR: Information Systems and Operational Research, vol.2, 1976, pp. 139–143.

[22] S.S. Heragu and A. Kusiak, "Machine layout problem in flexible manufacturing systems," Operations Research, vol. 2, 1988, pp. 258–268.

[23] M.F. Anjos and G. Yen, "Provably near-optimal solutions for very large single-row facility layout problems," Optimization Methods & Software, vol. 4-5, 2009, pp. 805–817.

[24] VIP PLANOPT Software, URL: `https://www.planopt.com/`. Accessed : Oct. 20, 2024

[25] M. Imam, and M. Mir, "Nonlinear programming approach to automated topology optimization, " ComputerAided Design, vol.2, 1989, pp. 107–115.

[26] M. Mir, and M. Imam, "Topology optimization of arbitrary-size blocks using a bivariate formulation, " Computer-Aided Design, vol.10, 1992, pp. 55–564.

[27] T. Dunker, G. Radons and E. Westkamper, "Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem," European Journal of Operational Research, vol. 1, 2005, pp. 55–69.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Dr.Aitzai Abdelhakim is the principal advisor, he conceived and designed the study. Dr. Lakehal Soumaya conducted the experiments, collected the data, and wrote the manuscript. All authors contributed to the manuscript revision and approved the final version of the manuscript.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflicts of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International , CC BY 4.0)

A Tables

A.1 Table 1

Table 1: Description of the used instances from literature

Problem	**Source**
SRFLP Instances	
LW5, LW11	[21]
S8, S10, S11	[1]
H20, H30	[22]
N25-01, N30-01	[20]
Sko64-01, Sko72-01	[23]
Sko81-01, Sko100-01	[23]
SR-CL Instances	
All instances	[22]
UA Instances	
L003	[24]
L004, L006	[5]
L008-IM	[25]
L010, L012	[5]
L0020-MI	[25]
L028	[26]
L050	[24]
L062	[27]
L100, L125-B	[24]

A.2 Table 2

Table 2: Computational results of the proposed method (SA-GA) compared to SA, GA, and EHSA for benchmark problems

Instance	**Best known**	**SA-GA**	**SA**	**GA**	**EHSA**
$N5-01$	15	15	15.23	15.41	15.23
$N6-01$	23	23	24.45	24.89	24.10
$N10-01$	62	62	66.01	65.85	63.15
$N15-01$	85	85	88.20	89.30	85.71
$N20-01$	159	159	166.12	164.05	160.75
Large Instances					
$N25-01$	4618	4618	8.45	4618	10.55
$N30-01$	8247	8247	10.81	8247	14.35
$Sko64-01$	96881	96945	23.23	96945	41.26
$Sko72-01$	139150	139243	30.17	139179	45.70
$Sko81-01$	205106	205233	35.09	205233	53.35
$Sko100-01$	378234	378376	57.50	378360	66.47

A.3 Table 3

Table 3: Computational results of SA-GA and EHSA to the SR-CL

Instances	$S^{\#}$	**SA-GA**$^{\#}$	**t(s)**	**EHSA**$^{\#}$	**t(s)**
$C5$	1.100	1.100	0.20	1.100	0.70
$C6$	1.990	1.990	0.69	1.990	1.29
$C7$	4.730	4.730	0.35	4.730	1.80
$C8$	6.295	6.295	0.95	6.295	1.15
$C12$	23.365	23.365	1.25	23.365	3.52
$C15$	44.600	44.600	4.30	44.600	6.30
$C20$	119.710	119.155	7.34	119.155	11.34
$C30$	334.870	334.870	9.56	334.870	17.56

A.4 Table 4

Table 4: A comparison between the results obtained by SA-GA and EHSA with VIP-PLANOPT and MI for UAF

Problem	S_v	S_{MI}	**SA-GA**	**EHSA**
$L003$	270.00	270.00	270.00	270.00
$L004$	1510.00	1510.00	1510.00	1510.00
$L006$	3379.00	3314.80	3351.50	3351.50
$L008-IM$	692.50	689.50	676.50	690.50
$L010$	19162.00	19279.00	18774.00	18920.00
$L012$	43180.00	43271.00	42889.50	42630.50
$L020-MI$	1157.00	1199.50	1199.04	1198.05
$L028$	6447.25	7536.80	7331.50	7302.81
$L050$	78224.70	77504.00	77943.23	77897.77
$L062$	3996206.00	4778682.00	3970161.50	3970161.50
$L100$	538193.10	54030.00	559725.67	567049.35
$L125-B$	1084451.00	1099290.00	1130400.12	1130372.00