International Journal of Computer Integrated Manufacturing

ISSN: 0951-192X (Print) 1362-3052 (Online) Journal homepage: https://www.tandfonline.com/loi/tcim20

Single and multi-row layout design for flexible manufacturing systems

Allan Tubaileh & Jamal Siam

To cite this article: Allan Tubaileh & Jamal Siam (2017) Single and multi-row layout design for flexible manufacturing systems, International Journal of Computer Integrated Manufacturing, 30:12, 1316-1330, DOI: 10.1080/0951192X.2017.1314013

To link to this article: https://doi.org/10.1080/0951192X.2017.1314013

Published online: 20 Apr 2017.

International Journal of Computer Integrated Manufacturing, 2017
Vol. 30, No. 12, 1316–1330, https://doi.org/10.1080/0951192X.2017.1314013

Single and multi-row layout design for flexible manufacturing systems

Allan Tubaileh[a]* and Jamal Siam[b]

[a]*Mechanical Engineering department, Birzeit University, Birzeit, Palestinian Territories;* [b]*Electrical Engineering department, Birzeit University, Birzeit, Palestinian Territories*

(*Received 2 April 2015; accepted 12 March 2017*)

In this paper, machines are allocated in horizontal rows along sides of an automated guided vehicle path so that the total cost of material transportation between machines is optimised. The problems of locating machines in single, double and multi-row layouts are addressed. Different layout arrangements as well as random permutations of machines are obtained using a simple construction algorithm, then the search for optimal solution and the best machine arrangements is obtained by implementing both Ant Colony and Simulated Annealing algorithms. Computational test examples show that the proposed method provides the best-known solutions for the single-row and double-row layout problems. Furthermore, experimental results demonstrate that both algorithms provide identical solutions for the single and multi-row layout problems.

Keywords: machine layout; ant colony optimisation (ACO); simulated annealing (SA) algorithm; flexible manufacturing systems (FMSs)

1. Introduction

An efficient layout design is crucial for flexible manufacturing systems (FMSs) implementation, because the layout is difficult to design, costly to modify and therefore significantly affects the efficiency of the entire system. The facility layout consists of three phases: layout design, material-handling system and facility system design (Tompkins et al. 1996). The material-handling system has a significant effect on the productivity of the entire system. It has been estimated that up to half of the total operating expenses within the manufacturing operation are due to material handling. Tompkins et al. (1996) have reported that effective layout design will reduce these costs by, at least, 10–30%. Therefore, in order to improve the production system, it is of great importance to develop the layout considering the material-handling system performance in the early stages of system design.

Several layout patterns are used in FMS. In these layouts, facilities are located in single-row, multi-row, circular loop. Since single-row and multi-row machine layouts are widely implemented in today's factories, therefore, the work of this paper will concentrate on these types of layouts.

In the single-row layout problem (SRLP), the machines are arranged along a single row on one side of the material-handling system path. The multi-row layout problem (MRLP) locates machines in multi-rows, where machines are laid on both sides of the material-handling system corridor. From practical point of view, MRLP layout often deals with more efficient and realistic material flow structure than SRLP. In actual facilities, rooms are arranged in both sides of a corridor and tool workstations are laid in the same manner for efficient space utilisation and minimum material-handling cost. For practical implementations of these layouts, one can refer to the work of Chung and Tanchoco (2010). Some researchers have discussed the advantages of MRLP compared to SRLP (e.g. Heragu and Alfa 1992; Solimanpur, Vrat, and Shankar 2005), but MRLP has not received much attention in literature.

In order to design an efficient manufacturing system, an appropriate material-handling device must be used for a specific layout configuration. Meanwhile, to increase the flexibility of the system, an automated handling device must be used for material transportation between machines such as automated guided vehicle (AGV) or a gantry robot. AGVs are very flexible, and they have better performance in moving in straight lines. Therefore, they are widely used in modern factories, especially FMSs in which machines are located in straight rows.

The machine layout problem in FMS involves the positioning of machines in single or multi-rows, so as to minimise the total material-handling cost. Machines are represented as rectangular shapes of unequal dimensions, in which the pick/up and drop-off points are located on one side of the rectangle. The pick-up/drop-off points are always directed towards the path of the AGV.

The problems presented in this paper are of combinatorial nature. The most popular meta-heuristic algorithms proposed to solve this type of problems are genetic algorithm (GA), simulated annealing (SA), tabu search (TS) and ant colony (AC). Recently, numerous studies were

*Corresponding author. Email: tubaileh@birzeit.edu

mainly concerned with performing comparative studies among these algorithms with respect to quality of the solution and the number of iterations required to obtain the best solution. In this context, Aarts and Lenstra (1997) presented a survey of these algorithms to the travelling salesman problem (TSP). In this study, the authors declared that if minimum cost is required and a considerable computational time is available, both SA and GA provide better solutions than could be obtained from other algorithms. Similarly, Alhamdy, Noudehi and Majdara (2012) presented a comparative study of the most popular meta-heuristic algorithms including SA, GA, TS and AC for the TSP. The results of this study showed that the AC provided the best performance and SA came in second place with little difference. The same authors indicated that GA is the most inefficient algorithm for TSP as it yielded the worst solution among other algorithms. On the other hand, Yossef, Sait and Adiche (2001) affirmed that all mentioned algorithms are efficient and robust to solve difficult combinatorial problems. Also, they concluded that each one has its own merits and it would be unwise to generalise the results obtained for a specific problem to other types of problems.

A recent work by Ahnon et al. (2014) implemented TS and SA algorithms for the corridor allocation problem (CAP). The authors pointed out that both algorithms were able to obtain the best-known optimal solutions for the benchmark problems considered by Amaral (2012); however, the SA presented the optimal solution in a less computational time. In addition, Chen et al. (2016) demonstrated that SA outperformed TS for obtaining solution of manufacturing cell planning. For all the motives discussed above, SA and ACO were used to solve the layout problems addressed in this paper.

This paper makes the following contribution. First, it presents a constructive algorithm that can be applied to solve SRLP, DRLP and MRLP. Second, the presented method considers the corridor length and width. It was verified through some numerical examples that the length of the corridor can considerably change the layout configuration. Third, the problem is formulated as an unconstrained layout problem which is solved by SA and ant colony optimisation (ACO). These algorithms have proven to be efficient for solving difficult combinatorial problems. Finally, experimental results show that the proposed technique provides good performance and the best-known solution for some problems solved by other methods.

The remainder of this paper is organised as follows. Section 2 reviews the previously proposed methods for the SRLP and MRLP, while Section 3 provides a description of the problem. The solution methodology implemented to search for the optimum solution is explained in Section 4. Section 5 presents a comparison of the computational performance of the proposed approach with other layout methods. Finally, conclusions and recommendations for future research are discussed in Section 6.

2. Literature review

The layout design of FMS should take into account various design aspects that are not extensively considered by the general facility layout problem (FLP), like machine's shape, pick-up/drop-off points and the material-handling device performance characteristics. Furthermore, machines are normally of unequal area, this will increase the complexity of the problem, mainly because unequal-area layout will add a constraint to the problem (Heng and Love 2000). Due to the differences between layout of FMS and the general FLP, the methods developed to solve FLP cannot be directly implemented for layout of FMS.

Traditionally, general FLP is solved as quadratic assignment problem (QAP). It is well known that the QAP is non-deterministic polynomia (NP)-hard and cannot be used to solve large-scale problems (Heragu and Alfa 1992). It has been demonstrated by many authors that QAP is an inappropriate method to be used for layout of FMS, since it cannot deal with machine dimensions and path of material-handling system. The SRLP and MRLP are classified as NP-hard, the application of exact methods to large-scale problem is time-consuming, and therefore various heuristics have been developed to obtain near optimal solutions for these problems. The heuristics developed in past years for SRLP and MRLP can be classified as

- Construction algorithm: This algorithm solves the layout from scratch and iteratively places one or a few machines at a time.
- Improvement heuristics: Using this type of algorithm, the problem is solved starting with an initial solution; then, the solution is improved during the algorithm evolution process. The initial solution could use constructive algorithm or random feasible solution.
- Meta-heuristic: GA, SA algorithm, ACO and TS are examples of meta-heuristics. These algorithms start with initial solution at each iteration, the solution may be obtained by a constructive-based algorithm or may be generated randomly. The meta-heuristic algorithm searches for the optimum solution among all possible solutions.
- Exact solution: Mixed integer problem (MIP) formulation was used by many researches in an attempt to find exact solution to SRLP and MRLP.

The SRLP is concerned with the placement of a number of facilities along one side of the corridor such that the material transportation cost between facilities is optimised. SRLP can be solved by one of these approaches discussed previously. For example, Kumar et al. (2008) presented a heuristic method that includes a scatter search for the SRLP.

A hybrid SA algorithm was presented by Heragu and Alfa (1992) for SRLP and MRLP of FMS. Samarghandi and Eshghi (2010) developed a TS for SRLP. Datta, Amaral and Figuiera (2011) developed a permutation-based GA which proved to give best solution for large-scale SRLP. Solimanpur, Vrat and Shankar (2005) solved the SRLP using ant colony system. A particle swarm optimisation was developed by Samarghandi, Taabayan and Jahantigh (2010) for the SRLP.

Amaral (2006) developed a strong MIP formulation which gives exact solutions to 15 machines by applying the linear-ordering polytope. Computational tests showed that his model provided better solutions than other previous MIP models proposed for the problem. Amaral (2009) presented a strong lower bound based on linear programming for large instances of SRLP. Similarly, a strong lower bound obtained via semi-definite programming was developed by Anjos, Kinning and Vannelli (2005).

Chung and Tanchoco (2010) addressed the double-row layout problem (DRLP) in which machines are placed in both sides of the corridor. A MIP model was presented with five heuristics in order to provide a good initial solution and corresponding upper bound of DRLP. However, the MIP model of Amaral (2013) was proven to give better performance than the model of Chung and Tanchoco (2010). As pointed out by Amaral (2013), problems size up to 12 machines were solved by both methods; however, larger size problems require high execution time. The DRLP was also addressed by Amaral (2012), denoted by CAP. He presented a MIP formulation and two heuristics procedures for large problems using data from benchmark SRLP. Murray, Smith and Zhang (2013) presented a method that combined combinatorial optimisation using local search with continuous optimisation using linear programming to solve the DRLP. Similarly, a multi-objective TS with linear programming approach was proposed by Zuo, Murray and Smith (2014) for an extended DRLP, in which the objective was to optimise the material transportation cost and layout area.

On the other hand, the MRLP consists of locating machines in different rows such that the total transportation cost between machines is optimised. MRLP caught the attention of a few researchers, for instance, Heragu and Alfa (1992) solved this problem using hybrid SA heuristic with machines of equal-area. In their work, the initial solution is generated by modified penalty algorithm in each iteration; then, the best solution is obtained by SA algorithm. The MRLP with machines of unequal-area was introduced by Gen, Ida and Cheng (1995). Their formulation considered machine dimensions and the fuzzy clearance between machines located in two rows. The permutation of machines in each row and neat clearance between machines and between boundaries is generated randomly; subsequently, a GA is implemented to find the best solution. It was assumed that the manufacturing facility is formed of two rows only where machines are allocated in one side of the corridor, also the corridor width was ignored by their formulation. MRLP of FMSs which employs AGV to feed machines allocated in one side of the corridor was presented by Ficko, Brezocnik and Balic (2004). In their procedure, machine sequences in each row and the favourable number of rows were established by GA. Unlike previous methods in which machines are located on one side of the corridor, locating machines on both sides of the corridor will considerably decrease the cost of material transportation as it will be illustrated later in this paper.

Tubaileh (2014) presented a new approach to address the single and double-row machine layout in FMS considering the kinematic constraints of the material-handling system. The objective of this paper was to optimise the time required by the autonomous vehicle to transport the material between machines taking into account the velocity, acceleration and kinematic constraints of the steering mechanism of the vehicle.

3. Procedure overview

3.1. Problem description

This paper investigates the single-row and multi-row machine layout in FMS. Machines are located along the corridor of one AGV which is used to transport material between machines within the same row and between different rows. The length of single row path is not limited, while locations of machines in multi-row system are limited by the vertical and horizontal dimensions of the facility. Machines are assumed to be rectangular in shape with pick-up/drop-off points located on the centre of either machine sides. The pick-up/drop-off point must be located on the side facing the corridor, so for simplicity, the horizontal dimension of the ith machine on which the pick-up/drop-off point is located is denoted as h_i and the other dimension is called vertical length v_i, no matter which is the longest one (see Figure 1).

The rectilinear corridor represents the flow path of material handling between machines. The objective is to minimise the cost of material movement, which is computed as the material flow times the distance measured a long rectilinear path. In single-row layout, the distance between machines is the absolute difference between the horizontal coordinates of pick-up/drop-off points as shown in Figure 1.

While, for the multi-row layout, the objective is to minimise the sum of the absolute horizontal and vertical distances between all pairs of machines, it should be taken

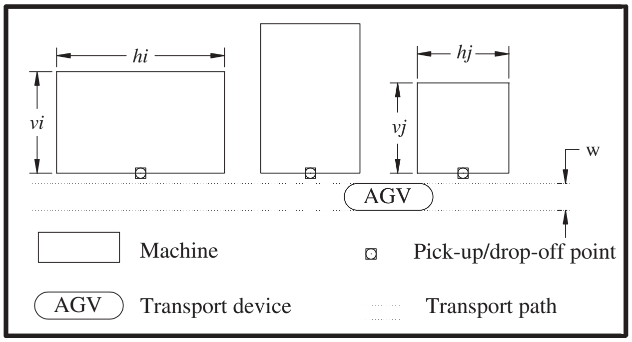

Figure 1. Representation of single-row machine layout in FMS.

into consideration that this distance must be the minimum, since there are two possibilities to turn from one row to the other as illustrated in the next section.

In solving the single and multi-row layouts, it is necessary to know the dimensions of the machines and the minimum allowable clearance distance between each pair of machines. Furthermore, flow between all pairs of machines must be provided during a certain period of time. Concerning the MRLP, the available space for the FMS (*hl, vl*) and the width of the AGV path (*w*) must also be known. In order to simplify the problem formulation, the clearance distance between machines is incorporated in the dimension of machines. Figure 2 describes a manufacturing system of four rows with an AGV used to transport material between machines located in different rows.

3.2. Representation

The solution procedure is divided into two basic steps: The first is concerned with determining the sequence and locations of the machines in the row, while the second tries to find the number of rows required for assigning all machines within the manufacturing system. The procedure iterates between these two steps until the best layout configuration is obtained. A SA algorithm, described in Section 4, is used to search for the best solution.

Assigning machines to rows starts from the lower left end, of the lowermost row, with the placement sequence following a random permutation of machines obtained by each iteration. When the total length of the lower side of the row exceeds the length of the row *a*, or the total length of machines exceeds the horizontal limit of the system, the last placed machine must be moved to the upper left end of the same path. Figure 3(a) illustrates this procedure, where machines 1, 2 and 3 are the first three cells to be placed. When attempting to allocate machine 4 in the lower row, it will not fit because either the pick-up/drop-off point position exceeds the row length (*a*) or the total machines length exceeds the limit of the facility *hl*. Then, machine 4 will be moved to the upper left side of the first corridor.

When the upper side of the first path (i.e. second row) is filled with machines 4 and 5 as shown in Figure 3(a), then the procedure proceeds to assign machine 6 to the left side of the next path (i.e. third row). In starting the third row located on the second path, two layout configurations might be generated according to the total number of rows required to locate all machines. These two layout configurations are shown in Figure 3, where the numbered machines indicate the order of placement. When the total number of rows is odd, the priority is to place machines in the upper side of next corridor. As explained in Figure 3 (a), this scheme will lead the proposed method to optimise the displacement of the AGV by reducing the vertical position of the row. However, for even number of rows, the assignment starts from the lower left side of the corridor until the row is filled. Then, the procedure continues to place machine 9 to the upper left end of the fourth row as shown in Figure 3(b).

For a given random permutation of machines for single-row layout, the placement starts from the left row end. An example of single-row layout is described in Figure 4 where the sequence of machines is given by $\pi = [4,1,3,2,5]$, and location of machines is denoted by $m = [m_1,m_2,m_3,m_4,m_5]$. Hence, m_1 corresponds to the location of machine 4 and m_2 represents location of machine 1, etc.

Regarding the multi-row layout, the procedure of calculating the coordinates of pick-up/drop-off points of machines is illustrated in Figure 5. The width of the transportation device and dimensions of machines must be taken into consideration when calculating the coordinates of the pick-up/drop-off points. As mentioned before, the minimum clearance between pairs of machines is included in the machine dimensions.

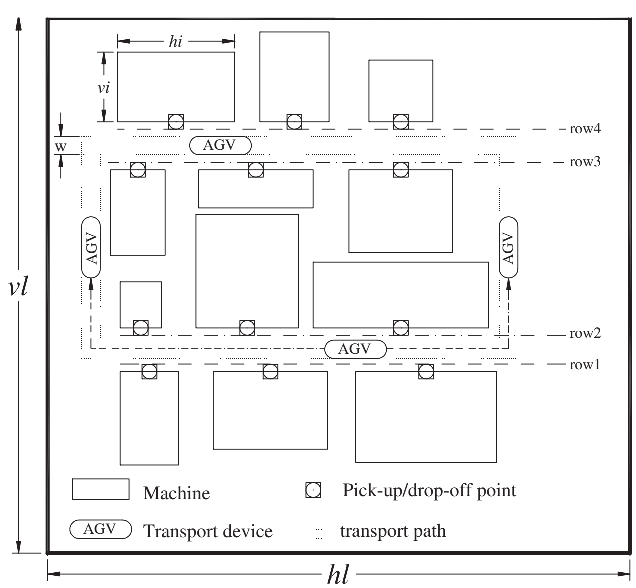

Figure 2. Representation of multi-row layout.

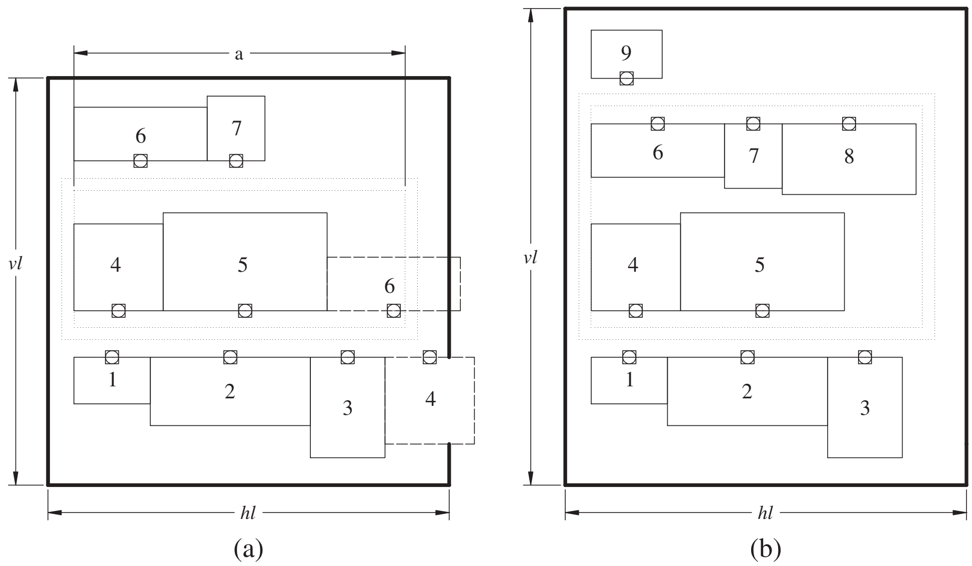

Figure 3. Spatial placement of machines for multi-row layout.

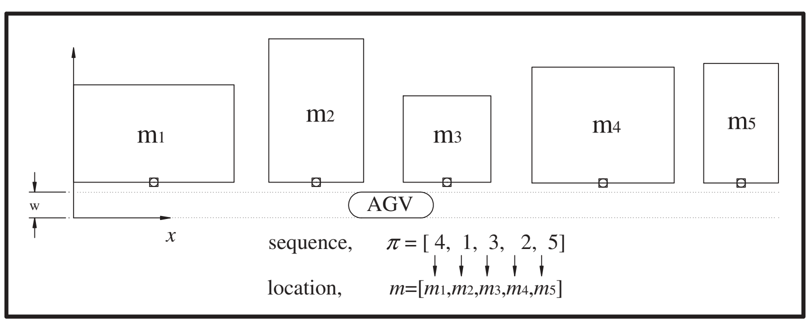

Figure 4. Spatial sequence of single-row layout.

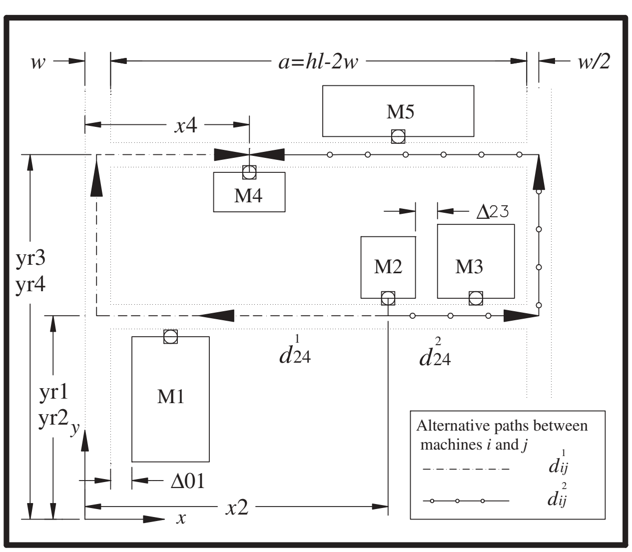

Figure 5. Flow path determination of multi-row arrangement.

According to Figure 5, machine locations are represented in the two-dimensional space by the following three variables: $x_{\pi_r(j)}, y_{\pi_r(j)}$ and $r = [1, 2, \ldots, R]$, where, $x_{\pi_r(j)}$and $y_{\pi_r(j)}$ represent the location of the pick-up/drop-off point of the jth machine in row $r \in R$, for a given permutation of machines π.

Denote by Π_n the set of all machines permutation π of $N = [1, \ldots, n]$, the total number of rows is denoted by R and n_r is the number of machines in the rth row such that $\sum_{r=1}^{R} n_r = n$. Then, the coordinates of machines of pick-up/drop-off points are given as follows:

$$x_{\pi_r(j)} = \frac{h_{\pi_r(j)}}{2} + \sum_{i=1}^{j-1} h_{\pi_r(i)}, \qquad r \in \{1, \ldots, R\}; \qquad (1)$$
$$1 \le j \le n; \quad \pi \in \Pi_n$$

$$y_{\pi_1} = y_{\pi_2} = \max\{v_{\pi_1(j)}\}, \qquad 1 \le j \le n_1 \tag{2}$$

$$y_{\pi_r} = y_{\pi_{r-1}} + \max\{v_{\pi_{r-1}(j)}\}, \quad n_{r-2} + 1 \le j \le n_{r-1}; \\ \text{for } R = 3, 5, \ldots, \text{odd}$$

$$y_{\pi_r} = y_{\pi_{r-1}} + \max\{v_{\pi_{r-1}(i)} + v_{\pi_r(j)}\}, \\ n_{r-2} + 1 \le i \le n_{r-1}; \quad n_{r-1} + 1 \le j \le n_r$$

$$y_{\pi_r} = y_{\pi_{r-1}} + \max\{v_{\pi_{r-1}(i)} + v_{\pi_r(j)}\}, \\ n_{r-2} + 1 \le i \le n_{r-1}; \quad n_{r-1} + 1 \le j \le n_r; \\ \text{for } R = 4, 6, \ldots, \text{even} \tag{3}$$

where $h_{\pi_r(j)}$ denotes the horizontal dimension of jth machine on row r for a given machine permutation π and y_{π_r} represents the y coordinates of row $r \in R$. Equation 1 expresses the x coordinates of machine pick-up/drop-off points, also it prevents machines overlapping. In order to reduce the complexity of the problem formulation, the clearance required for any machine can be easily included in its dimensions. As described by Equation 2, the y coordinate of the first corridor is defined by the maximum vertical dimension (v_j) of the machines located in the first row, while Equation 3 represents the y coordinate of the other corridors, which is determined by the sum of the maximum height of machines in the last row and the y coordinate of the previous row, for odd number of rows. On the other hand, for even number of rows, the coordinates are defined by the distance of the previous corridor and the maximum height of the machines in the last two rows.

Once the coordinates of pick-up/drop-off points are calculated as explained in Figure 5, the distance travelled by the material-handling device between a pair of machines can be determined. If several paths between pair of machines are possible, the shorter one is selected (Ficko, Brezocnik, and Balic 2004). If machines i and j are located in the same row, the distance of the path d_{ij} is calculated according to the following equation:

$$d_{ij}^{\pi} = |x_{\pi_r(i)} - x_{\pi_r(j)}| \tag{4}$$

When machines are located in different rows, there are two possible routes, as shown in Figure 5. The distance associated with these two possible routes is defined by the following equations:

$$d_{ij}^{\pi 1} = x_{\pi_r(i)} + x_{\pi_r(j)} + w + |y_{\pi r(i)} - y_{\pi r(j)}| \tag{5}$$

and

$$d_{ij}^{\pi 2} = (a - x_{\pi_r(i)}) + (a - x_{\pi_r(j)}) + w \\ + |y_{\pi r(i)} - y_{\pi r(j)}| \tag{6}$$

Between these two possible routes, the minimum path length d_{ij}^{π} is selected as indicated by the following equation:

$$d_{ij}^{\pi} = \min(d_{ij}^{\pi 1}, d_{ij}^{\pi 2}) \tag{7}$$

The same procedure is used to determine the minimum path lengths d_{ij} between all pairs of machines. Accordingly, to solve the SRLP and MRLP, it is necessary to know the material flow between all pairs of machines during a certain period of time, as well as the cost of travel which depends on the material-handling system.

Taking into consideration all these issues, the machine layout problem in FMS can be addressed as follows: for a given permutation of machines, π, find the optimum locations of n machines such that the total cost of travel between all pairs of machines is optimised. Thus, the objective function can be formulated as

$$z = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} f_{ij} d_{ij}^{\pi}, \qquad \pi \in \Pi_n \tag{8}$$

where f_{ij} and c_{ij} denote the number of trips and the material transportation cost between machines i and j, respectively, while d_{ij}^{π} is the length of the path between machines i and j corresponding to permutation π, as given by Equation (7).

Equation 8 represents the sum of the transportation cost between machines for a given permutation π. To find the optimum cost function for all possible permutations of machines, several meta-heuristics can be implemented like GA, TS, SA and ACO. In this paper, the SA and ACO algorithms have been utilised to find the best solution among all randomly generated layout problems for both SRLP and MRLP.

4. Solution methodology

The problems discussed in the previous section are of combinatorial nature, at which the meta-heuristic algorithms were proven to be effective in solving similar problems. A brief description of SA and ACO implemented to solve the problems addressed in this paper will be presented in the following sections.

4.1. Simulated annealing algorithm

SA is a general optimisation procedure proposed to solve large combinatorial problems based on randomisation technique. Kirkpatrick, Gelatt and Vecchi (1983) introduced the concept of SA inspired by the physical annealing process of solids. Solution of combinatorial problems is similar to the state of solid and the cost of the solution is equivalent to the energy of state.

The working principle of SA is simple. During the search process, the algorithm tends to ‘hill climb’ towards a better solution, while a worse solution is accepted with a certain probability (P).

This attribute enables SA to obtain high-quality solution by permitting it to escape from local optimum.

The SA algorithm consists of four basic components:

(1) Configurations: all possible solutions for combinatorial problems (states).
(2) Configuration change: a mechanism to change the configuration by small perturbation of state.
(3) Cost function: a measure of the quality of solution.
(4) Cooling schedule: reduce the temperature to produce a final equilibrium state with minimum energy.

The general scheme of the SA procedure can be described by the following pseudo-code programme:

1. Initialise
   Initial temperature (T_o), initial configuration (S).
   2. While the stop criterion is not satisfied
   2.1. Perform the loop (*NOVER*) times
      2.1.1. Generate random neighbour S' from S
      2.1.2. Let $\Delta = \text{cost}(S')\text{-cost}(S)$
      2.1.3. If $\Delta \leq 0$, then set $S = S'$
      2.1.4. If $\Delta > 0$, then set $S = S'$ with probability $p = \exp^{(-\Delta/T)}$
      2.2. Set $T_{k+1} = r\ T_k$, where r is the reduction ratio ($0 < r < 1$).
   3. Go to 2.

In this work, the generation mechanism described by step 2.1.1 randomly selects two machines and exchanges the locations of these machines. The performance of the SA strongly depends on the selection of a set of critical values of the control parameters. Particularly, the parameters that must be carefully chosen for finite-time performance are described in the following sections.

4.1.1. Cooling schedule

Evidently, the cooling schedule includes the initial temperature, temperature reduction formula, the number of iteration at each temperature and the time of the annealing process.

The initial temperature (T_o) must be high enough in order to make sure that the search will reach all possible states. In physical analogy, the solid is heated up to high temperature until all particles are randomly arranged in the liquid phase. It was stated before that the quality of the solution strongly depends on the initial temperature value. Hence, the value of the initial temperature is crucial. The SA starts with initial probability P_o, from which the initial temperature T_o is determined according to the Metropolis criterion, $P_o \cong \exp^{(-\bar{\Delta}/T_o)}$, while the mean cost increase $\bar{\Delta}$ of the transitions was calculated by running the programme several pilot runs.

SA technique requires very slow cooling schedule to guarantee good performance. Slow cooling process requires high computational time. Hence, the performance of a finite-time implementation strongly depends on the choice of control parameters. The CPU time is affected by the number of configuration changes at each temperature which is also identified as Markov chain length.

When the number of configuration reached the upper bound of the Markov chain length, the temperature is reduced according to a cooling schedule $T_{k+1} = rT_k$, where r is the cooling ratio ($0 < r < 1$). Hence, the equilibrium state will be attained when the final temperature (T_f) is reached.

The optimisation process stops when the optimal configuration remains unchanged for a given number of reduction stages (*NOVER*) or when the number of successful transitions exceeds a prescribed value (*NLIMIT*).

A flow chart of the proposed SA is shown in Figure 6.

4.1.2. Parameters selection for SA

Several tests have been performed to select the appropriate parameters required to optimise the layout problem using the SA algorithm. The parameters provided by previous researches are first selected; then, a change of one parameter at a time is performed to obtain the best performance of the SA. For example, the parameters *NOVER* and *NLIMIT* are set proportional to the number of machines n at ($100 \times n$) and ($10 \times n$), respectively, as indicated by Heragu and Alfa (1992). The same authors recommended to take the temperature reduction factor r from the range 0.8 to 0.99. A value of 0.9 is selected in this research for an optimum layout convergence in a reasonable execution time.

4.2. ACO

ACO is a meta-heuristic algorithm designed to solve combinatorial optimisation problems. ACO principles are based on the natural behaviour of ants while searching for the shortest path between their nest and some food source. Ants need to travel several times between their nest and a food source; while travelling, they leave in the path a chemical substance called pheromone. The path with high concentration of pheromone has the high probability to be selected by the ants to travel and hence more pheromone is deposited through that path. Dorigo, Maniezzo and Colorni (1996) developed the first version of ACO to solve the TSP.

Generally, the optimisation problem under consideration is transformed into a weighted Graph. Then,

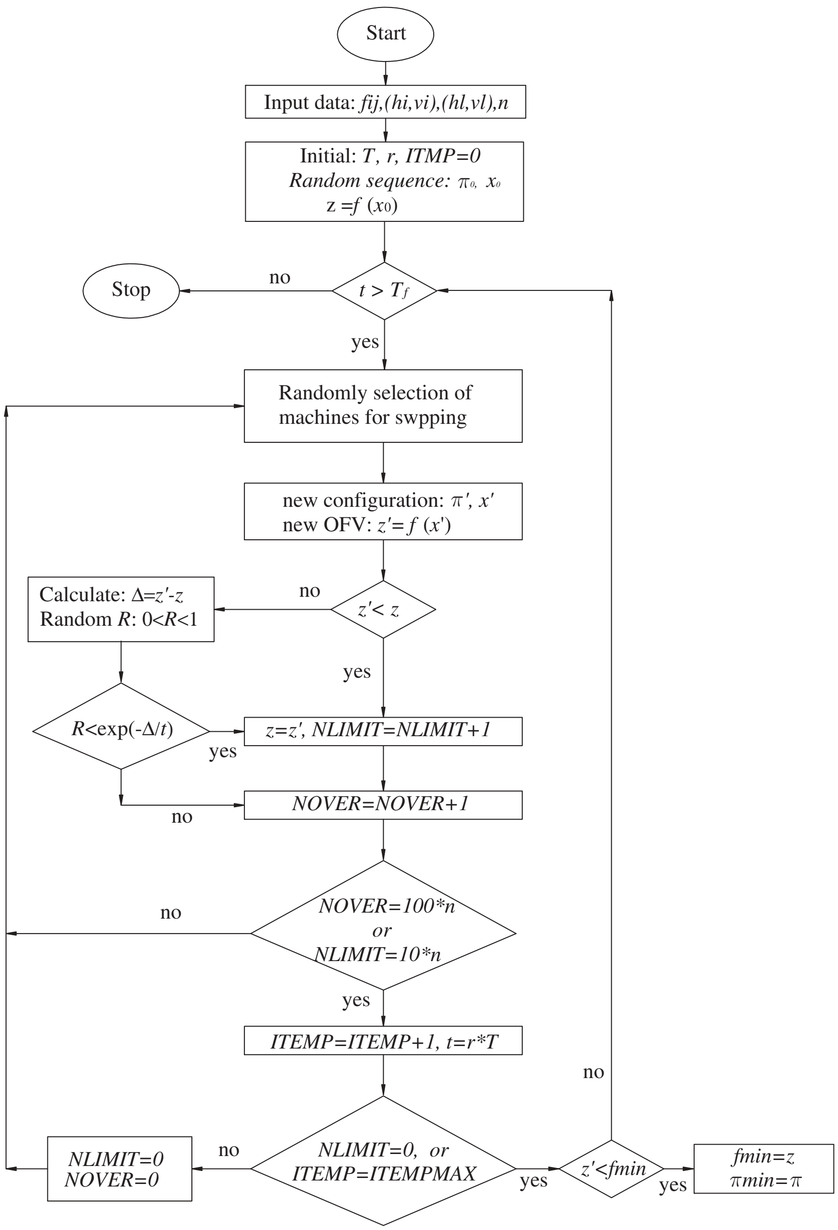

Figure 6. Flow chart for optimum layout search by SA.

the ACO system iteratively dispenses a group of ants onto the graph to create tours corresponding to possible solutions. The optimisation technique by ACO has two important phases, the tour creation and the pheromone-updating scheme. The ant's move from a node to another in the network is probabilistically selected based on pheromone quantities deposited on the tour from the current node to the next node in the tour. Since all ants start from the same node, a feasible solution is obtained when all ants have been selected. When all ants have developed their corresponding solutions, the pheromone trails are updated. The pheromone values are decreased through evaporation for tour with poorer solution, on the other side; pheromone values are increased for the tours which produce the best solutions. The process of solutions development and pheromone updating is repeated until the stopping condition has been reached.

4.2.1. AC approach for the machine layout problem

The problem representation is the most important decision to be made in solving any problem by ACO since inappropriate representation will lead to poor solution quality. The ACO algorithm for the machine layout problem will be illustrated in the following steps:

(1) The layout problem is transformed into a directed graph, a set of n nodes and a set of m arcs (i,j). The number of machines will be represented by the number of nodes in the graph and the number of nodes will be the number of ants.
(2) An ant is positioned arbitrarily to a node; then, an equal pheromone is dispensed to each node. The initial pheromone is given by the following equation:

$$\tau_o = \frac{1}{n \times \sum_{i \in C} w_i} \tag{9}$$

where C is the initial machines sequence (machines permutation), n is the number of nodes in the initial sequence and $\sum_{i \in C} w_i$ is the summation of the weights in the initial sequence.

(3) The ant k in node i will select the next node to visit j according to the following probabilistic equation:

$$p_j^k = \begin{cases} 1, & \text{if } q < q_o \text{ and } j = \arg\max\left\{\tau_r \times \eta_{rk}^{\beta}\right\}_{r \in A_k} \\ 0, & \text{if } q < q_o \text{ and } j \neq \arg\max\left\{\tau_r \times \eta_{rk}^{\beta}\right\}_{r \in A_k} \\ \frac{\tau_r \times \eta_{jk}^{\beta}}{\sum_{r \in A_k} \tau_r \times \eta_{rk}^{\beta}}, & q > q_o \end{cases} \tag{10}$$

where A_k is the set of all possible nodes to be visited by node k and the variable η_{jk} is given by

$$\eta_{jk} = \frac{\sum_{(i,j) \in E} \psi_k(r,j)}{w(j)} \tag{11}$$

where η_{jk} stands for the number of nodes joined to node j and it must be visited by ant k.

(4) When an ant selects a node, the pheromone is updated on the node as follows:

$$\tau_i = (1-\rho)\tau_i + \rho\tau_o \tag{12}$$

where ρ represents the pheromone evaporation rate $\rho \in [0,1]$ and τ_o is the initial pheromone defined by Equation (9).

(5) The value of the objective function is calculated after each ant has completed its tour. The objective value for a given tour is saved as local best.
(6) Steps 3 and 4 are repeated till all ants have completed their tours. The lowest objective function obtained by the current iteration is stored as best solution.

(7) Let S be the best solution found by the current iteration. The pheromone left on node i is updated according to the following rule:

$$\tau_i = \begin{cases} (1-\rho)\tau_i + \rho\Delta\tau_i, & \text{if } i \in S \\ (1-\rho)\tau_i, & \text{otherwise} \end{cases} \tag{13}$$

where $\Delta\tau_i$ represents the pheromone quantity to be deposited and is given by

$$\Delta\tau_i = \frac{1}{\sum_{j \in S} w_j} \tag{14}$$

(8) Steps 3–7 are repeated until the maximum number of iterations is reached.

4.2.2. *ACO parameters selection*

The parameters are tested several times in order to select the values which will give the best solutions. In our simulation examples, the ACO parameters are set as

$\rho = 0.9$, $\beta = 2$, $q_o = 0.9$, number of ants = number of machines.

5. Computational results

Three sets of experimental analysis were conducted in order to study the effectiveness of the proposed method. The first set has focused on comparing the results of single-row layout obtained by the proposed method with those obtained by others researchers. The second investigation set compared the proposed method applied to double-row layout with the methods of Amaral (2012), Das (1993) and Ahonen et al. (2014). Finally, the third study explored the proposed new approach for solving the multi-row layout using SA and ACO techniques. The proposed algorithms are solved using Matlab 7.9 and are executed on an HP notebook having 2.0 GHz Intel Centrino dual-core and 3 GB of RAM.

In order to investigate the effectiveness of the proposed method, several traditional problems of SRLP with known best solutions are considered. Table 1 shows these problems with the corresponding average transportation cost function and the best-known optimal solution obtained by previous works (Datta, Amaral, and Figuiera 2011; Amaral 2009; Samarghandi, Taabayan, and Jahantigh 2010; Samarghandi and Eshghi 2010). It must be declared that instances for the SRLPs are adopted from Amaral (2009) and Anjos and Vannelli (2008). It can be observed from Table 1 that the SA algorithm was able to obtain the optimal solution for small-size SRLP in nearly each run (out of 50 runs). In contrast, the performance of the SA becomes dependent on its various parameters as the problem size is increased, as observed from the standard deviation values of Table 1.

Solimanpur, Vrat and Shankar (2005) stated that the computational time should not be considered for comparing the performance of different algorithms as the computed machines are different. Instead, the number of function evaluations would provide a better alternative for the estimation of computational performance for algorithms of meta-heuristic nature. As it can be noted from Table 1 that the proposed SA method was able to obtain the best solution in a reasonable number of function evaluations compared with the GA proposed by Datta, Amaral and Figuiera (2011) and particle swam optimisation (PSO) of Samarghandi, Taabayan and Jahantigh (2010) for SRLP. Table 2 shows the permutations of machines obtained by the proposed method, and it can be observed that they are identical to those obtained by all previous works which were able to achieve the best-known solution of the considered SRLPs.

Table 3 depicts the results of SRLP using ACO. It can be observed from Table 3 that ACO was able to produce the best-known solutions for the benchmark SRLPs. However, the computational time required by ACO was inferior to that required by SA approach.

In this paper, the DRLP denoted as SA-DRL is considered as a special case of MRLP. The performance of the proposed SA-DRL is displayed in Tables 4 and 5. The problems in Table 4 are adopted from Das (1993), while the problems in Table 5 are taken from Amaral (2012). As it can be observed from Table 4, the proposed SA-DRL provides a better solution in nearly all the tested problems solved by the Spline method presented by Das (1993) except for problem 1 which consists of only 4 machines. As the number of machines increases, the performance of SA-DRL becomes more efficient than the Spline method.

Table 2. The best permutations obtained by the proposed SA method for SRLP.

Problem number	Number of machines	Optimal OFV	Permutations of machines
1	4	638	3–4–1–2
2	5	151	3–4–5–1–2
3	8	801	6–8–3–5–1–2–7
	8-H	2324.5	7–8–1–5–4–6–3–2
4	9	2469.5	2–3–6–9–1–5–7–4–8
	9-H	4695.5	5–9–2–1–7–3–6–4–8
5	10	2781.5	8–6–2–4–10–5–7–1–3–9
6	11-S	6933.5	7–2–1–10–4–3–6–5–8–11
	11–LW	6933.5	11–10–3–8–2–7–5–1–9–4–6
7	15	6305	1–2–13–9–11–8–7–12–14–4–3–5–6–15–10
8	20	15,549	17–13–5–6–7–20–8–12–11–4–16–15–2–14–19–10–18–3–9
9	30	44,965	28–4–14–20–29–8–19–30–16–27–25–11–3–7–21–9–10–13–23–22–1–18–15–17–6–24–12–26–5–2

Furthermore, Das reported that the Spline algorithm was unable to reach optimal solutions for large size problems ($n > 12$), unlike the proposed method which can be implemented to solve large size problems in a reasonable computational time as it can be seen later in this section.

Amaral (2012) denoted the DRLP by as the CAP. He presented a MIP formulation and two heuristics to solve the CAP problems that are too large to be solved by exact approaches. The CPLEX 12.1.0 solved the MIP formulation to optimality with problems up to 13 machines. He also demonstrated that the exact formulation could not obtain an optimal solution for the DRLP with 15 facilities even after 8.6 h of execution time. The data from SRLP benchmark problems were solved by Amaral (2012) to test the

Table 1. Performance of the proposed SA method for traditional SRLP.

Problem number	Number of machines	Proposed SA method: Average OFV†	Proposed SA method: Standard deviation	Best-known OFV‡	Average time (s)	Average no. of function evaluations: Proposed SA	Average no. of function evaluations: Datta, Amaral, and Figuiera (2011)
1	4	638	0.0	638	0.2	99	–
2	5	151	0.0	151	0.4	140	50
3	8	801	0.0	801	2.10	185	–
	8-H	2324.5	0.0	2324.5	2.07	223	383
4	9	2469.5	0.0	2469.5	1.2	274	–
	9-H	4695.5	0.0	4695.5	1.43	421	
5	10	2781.5	0.0	2781.5	1.65	526	918
6	11-S	6933.5	0.0	6933.5	1.86	866	1096
	11-LW	6933.5	0.0	6933.5	1.81	905	–
7	15	6305	0.0	6305	16.30	1028	–
8	20	15,552	2.1	15,549	22.25	4230	4034
9	30	44,968	3.4	44,965	64.22	10,230	10,576

†OFV = Objective function value; ‡found by GA of Datta et al., PSO Samarghandi, Taabayan and Jahantigh (2010) and LP of Amaral (2009).

Table 3. Performance of the proposed ACO method for traditional SRLP.

Problem number	Number of machines	Proposed ACO method: Average OFV	Proposed ACO method: Standard deviation	Best-known OFV	Average time (s)	Permutations of machines
1	4	638	0.0	638	0.18	3–4–1–2
2	5	151	0.0	151	0.3	3–4–5–1–2
3	8	801	0.0	801	1.40	6–8–3–5–1–2–7
	8-H	2324.5	0.0	2324.5	1.87	7–8–1–5–4–6–3–2
4	9	2469.5	0.0	2469.5	0.9	2–3–6–9–1–5–7–4–8
	9-H	4695.5	0.0	4695.5	1.3	5–9–2–1–7–3–6–4–8
5	10	2781.5	0.0	2781.5	1. 5	8–6–2–4–10–5–7–1–3–9
6	11-S	6933.5	0.5	6933.5	1.66	7–2–1–10–4–3–6–5–8–11
	11-LW	6933.5	1.5	6933.5	1.72	11–10–3–8–2–7–5–1–9–4–6
7	15	6305	2.1	6305	8.30	1–2–13–9–11–8–7–12–14–4–3–5–6–15–10
8	20	16,415	2.4	15,549	12.5	5–6–17–13–7–20–10–12–11–4–16–15–2–14–19–8–18–3–9
9	30	45,054	3.7	44,965	14.9	21–25–11–20–15–17–19–30–10–28–4–14–3–7–21–9–16–13–1–22–23–18–29–8–6–24–12–26–5–2

performance of his proposed algorithms. The above problems are solved in this paper in order to compare the proposed SA-DRL with previous approaches developed for the DRLP. It can be observed from Table 5 that the proposed SA-DRL was able to attain identical solutions obtained by the CAP approach. Furthermore, the previous tests on DRLP show a significant discrepancy in computational time among the three approaches with the SA-DRL being significantly faster, especially for the large test problems. For instance, the SA-DRL requires an average computational time of about 70 s to solve problems with 30 machines, while the best CAP heuristic requires more than 2 h. This advantage will be confirmed in solving large-sized problems for the MRLP as illustrated later in this section.

It should be pointed out that Amaral (2012) presented two neighbourhood search-based heuristics to find the optimal solutions for large problems adapting the well-known 2-opt and 3-opt algorithms to swap machines positions. He also used a variable denoted t in order to split the layout in two rows (i.e. t = number of machines in row 1). Unlike Amaral's approach, this paper generates machine permutation randomly; the number of machines to swap can be specified by the user and this number could be reduced in each stage by using the same criteria applied for reducing the temperature of the SA algorithm. Furthermore, the number of machines in each row is defined by the length of the row, which is more reasonable from practical point of view than choosing the number of machines in each row.

A recent work by Ahnon et al. (2014) implemented a TS and a SA algorithms for the CAP. The performance of both algorithms were tested using the instances considered by Amaral (2012). The authors pointed out that both algorithms were able to obtain the best-known optimal solutions for the benchmark problems; however, the SA presented the optimal solution in a less computational time. Table 6 compares the results obtained by the proposed SA-DRL algorithm and those obtained by TS and SA implemented by Ahnon et al. (2014). It can be observed that the proposed algorithm was able to obtain the optimal solutions achieved by the SA and TS algorithms of Ahonen et al. (2014).

The multi-row layout approach was analysed by solving FMSs of 30 machines, and the performance of both the SA and ACO approaches is displayed in Tables 7 and 8, respectively. The average OFV of each problem is calculated for 50 runs. In order to prove the effectiveness of the proposed method, the MRLP was solved using the same problems originally handled as SRLP. Both algorithms have obtained the same objective value as can be shown in Tables 7 and 8. As for the SRLP, the ACO exhibited a better performance regarding the computational time required to solve the MRLP.

Figure 7(a) describes the placement procedure for a row length of 25 m at which the algorithm was capable to place all machines in three rows. In the third row, machines are

Table 4. Comparisons of material transportation cost of the SA-DRL with Spline-Layout of Das (1993).

Problem number	Number of machines	OFV: SA-DRL	OFV: Das-Spline†	SA/Spline benefit
1	4	819	676	‡
2	6	1890.5	2351.2	24.3%
3	8	7099.5	7877.5	10.9%
4	10	13,498	17,365	28.6%
5	12	37,018	44,876.7	21.2%

†Spline objective is calculated after excluding the offset distance o.
‡The objective of Spline method is better than SA-DRL.

Table 5. Comparison of material flow cost of the proposed SA-DRL with DRLP of Amaral (2012).

Problem number	Number of machines	Optimal OFV			Time (s)		
		SA-DRL	CAP	MIP	SA-DRL	CAP†	MIP
1	9	1181.5	1181.5	1181.5	1.5	1.3	18.2
	9H	2294.5	2294.5	2294.5	2.7	1.33	1145.94
2	10	1374.5	1374.5	1374.5	2.5	2.11	62.75
3	11	3439.5	3439.5	3439.5	6.1	3.0	496.2
4	12a	1529.0	1529.0	1529.0	9.3	4.99	1869.3
5	15	3195	–	–	14.5	–	–
	30-1	4155.0	4155.0	–	44.2	6070.5	–
6	30-2	10,779.5	10,779.5	–	60.9	7412.6	–
	30-3	22,702.0	22,702.0	–	102.4	7902.4	–

†Solutions obtained by heuristic-2 with 2-opt.

Table 6. Comparison of results obtained by SA-DRL with those obtained by SA and TS of Ahonen et al. (2014).

Problem number	Number of machines	SA-DRL		Ahnon et al. (2014)			
		OFV	Time (s)	SA	Time (s)	TS	Time (s)
1	9	1181.5	1.5	1181.5	4.05	1181.5	1.14
	9H	2294.5	2.7	2294.5	3.46	2294.5	1.15
2	10	1374.5	2.5	1374.5	4.88	1374.5	1.58
3	11	3439.5	6.1	3439.5	5.97	3439.5	2.02
4	12a	1529.0	9.3	1529.0	7.36	1529.0	2.43
5	15	3195	14.5	3195	12.2	3195	4.7
	30-1	4155.0	44.2	4155.0	4.41	4155.0	26.51
6	30-2	10,779.5	60.9	10,779.5	4.40	10,779.5	37.68
	30-3	22,702.0	102.4	22,702.0	4.45	22,702.0	35.33

located above the path of the AGV, as explained in the previous discussion. By reducing the width to 22 m, the layout configuration changes to four rows as shown in Figure 7(b). The assignment of machine starts on the lower side of the second path. When the third row is completely filled, machines are placed on top of the second path (fourth row) starting from the left side.

Figure 8 shows an efficient multi-row layouts of large-scale problems of 20 and 30 machines, problems 6 and 7 in Table 7. Obviously, Figure 8 shows how efficient is the present approach to develop layouts applicable to FMS served by AGV as a material-handling system. A zero clearance distance has been considered between machines in all the problems solved in this paper. Since, the problems reported in Table 7 were taken from SRLP benchmark data, the vertical dimension of machines (v_i) are not available in literature devoted to solve this type of problems. Hence, in order to assign machines in multi-row

Table 7. Performance of the proposed SA based approach for multi-row layout.

Problem number	Number of machines	Optimal value	SA objective value		Average time (s)	Average number of iterations	Max. row width	Number of rows
			Average	Standard deviation				
1	8	418	418	0.0	3.07	544	17	2
	8	569	569	0.0	2.04	420	10	4
2	9	1597.5	1597.5	0.0	2.67	845	20	3
3	10	1706.5	1706.5	0.0	2.86	1420	20	4
4	11	4206.5	4206.5	0.0	28.6	2892	25	3
	11	4980.5	4980.5	0.0	29.3	2863	22	4
5	15	4170	4171.8	3.5	36.9	5953	27	4
6	20	9119	9122.2	4.3	45.8	10,536	35	4
7	30	23,032	23,036.1	7.2	129.2	18,364	45	4

Table 8. Performance of the proposed ACO based approach for multi-row layout.

Problem number	Number of machines	Optimal value	ACO objective value Average	Standard deviation	Average time (s)	Average number of iterations	Max. row width	Number of rows
1	8	418	418	0.0	1.42	55	17	2
	8	569	569	0.0	1.88	56	10	4
2	9	1597.5	1597.5	0.0	2.17	88	20	3
3	10	1706.5	1706.5	0.0	2.46	97	20	4
4	11	4206.5	4206.5	0.8	16.5	105	25	3
	11	4980.5	4980.5	1.8	19.2	110	22	4
5	15	4170	4171.8	2.5	26.7	123	27	4
6	20	9119	9122.2	3.8	33.5	200	35	4
7	30	23,032	23,036.1	6.4	86.4	200	45	4

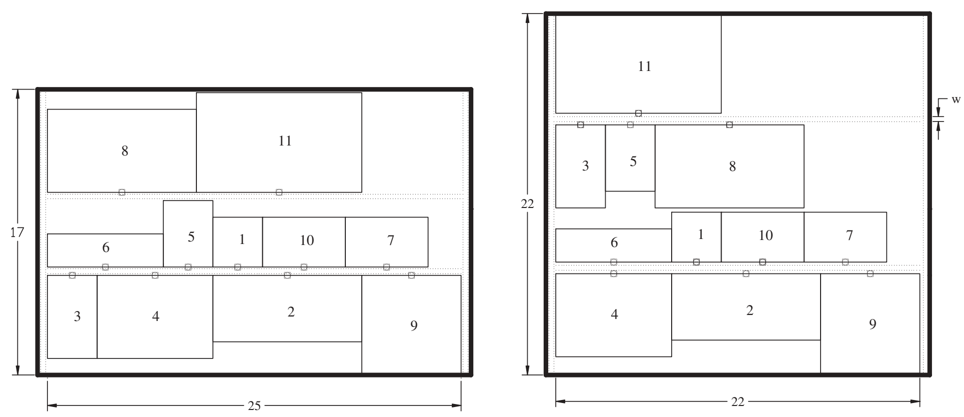

Figure 7. Layout configuration change according to the length of rows.

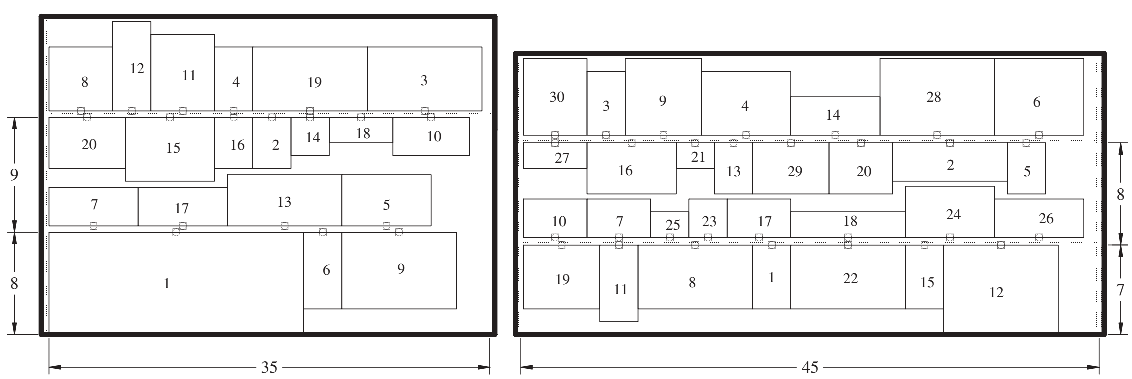

Figure 8. Layout of 20 and 30 machines.

layout, the width of the machines must be known. The vertical dimension of machines for the corresponding problems were assumed by the author and they are displayed in Table 9. Finally, Table 10 presents the optimum permutations of machines obtained for the multi-row layouts shown in Tables 7 and 8.

6. Conclusions

This paper presented a method to solve the machine layout problem in FMS which involves locating machines in rows along straight corridors such that the sum of material transportation cost between each pair of machines is minimised. A simple constructive approach is proposed to

Table 9. Machine vertical dimensions for the MRLP.

Problem number	No. of machines	Machine vertical dimension (v)
1	8	3, 3, 1, 3, 4, 4,5, 2
2	9	3, 4, 4, 5, 4, 4, 3, 5, 6
3	10	10, 12, 15, 9, 6, 25, 20, 5, 11, 13
4	11	3, 4, 5, 5, 4, 2, 3, 5, 6, 3, 6
5	15	8, 4, 5, 5, 4, 6, 3, 5, 6, 3, 6, 7, 4, 3, 5
6	20	8, 4, 5, 5, 4, 6, 3, 5, 6, 3, 6, 7, 4, 3, 5, 4, 3, 2, 5, 4
7	30	5, 4, 5, 5, 4, 6, 3, 5, 6, 3, 6, 7, 4, 3, 5, 4, 3, 2, 5, 4, 2, 5, 3, 4, 2, 3, 2,6, 4, 6

Table 10. Best permutations obtained by the proposed SA and ACO method for MRLP.

Problem	Number of machines	Optimal OFV	Permutations of machines
1	8	418	7–1–2–4–3–5–8–6
	8	574	5–7–3–1–2–8–4–6
2	9	1597.5	9–5–8–6–7–1–4–2–3
3	10	1706.5	5–4–2–8–10–6–7–1–3–9
4	11	4206.5	3–4–2–9–6–5–1–10–7–8–11
	11	4980.5	4–2–9–6–1–10–7–3–5–8–11
5	15	4170	2–8–11–12–9–13–14–7–5–4–3–10–15–1–6
6	20	9119	1–6–9–7–17–13–5–20–15–16–2–14–18–10–8–12–11–4–19–3
7	30	23,032	19–11–8–1–22–15–12–10–7–25–23–17–18–24–26–27–16–21–13–29–20–2–5–30–39–4–14–28–6

generate permutations of machines locations in straight rows. Using this technique, it was possible to treat the layout problem as unconstrained optimisation problem. Accordingly, the search for optimal solution is obtained by implementing a SA and AC algorithms, which have proven effective in solving similar problems of combinatorial nature. The proposed approach has the flexibility to solve single-row, double-row and multi-row layouts which are commonly implemented in recent FMSs. The numerical examples presented in this paper favourably compared to the best solutions found in literature for SRLP and DRLP. The proposed approach was able to obtain the optimum solutions reported by previous works for benchmark SRLP. For the DRLPs, the proposed method was also capable to produce the best-known solution obtained by previous works. Computational tests showed that both SA and ACO algorithms have been proven to be efficient in solving SRLPs, DRLPs and MRLPs, since both algorithms produced identical layouts for all tested problems. Unlike other methods that addressed the layout problem with a predetermined number of rows and a fixed distance between rows, the method presented in this paper was able to find the optimal number of rows and the distance between rows for a specific layout requirements.

In future work, the model may be extended to consider the row length in the objective function in order to obtain an optimum length of each row. Furthermore, the pick-up and drop-off points may be separated or located on different sides of machines, which will affect the layout configuration and the cost function. One of the most important issues that has not received attention in literature is the kinematical constraints of the material-handling device. In future research, the performance characteristics of the material-handling system should be considered in the early stages of the layout design.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Aarts, E. H., and J. K. Lenstra. 1997. *Local Search in Combinatorial Optimization*, 215–310. London: *John Wiley& Sons*.

Ahonen, H., A. G. Alvarenga, and A. R. S. Amaral. 2014. "Simulated Annealing and Tabu Search for Corridor Allocation Problem." *European Journal of Operational Research* 232: 221–233. doi:10.1016/j.ejor.2013.07.010.

Alhamdy, S., A. Noudehi, and M. Majdara. 2012. "Solving the Travelling Salesman Problem Using Ant Colony Algorithm Comparing with Tabu Search, Simulated Annealing and Generic Algorithm." *Journal of Applied Sc Researcher* 8 (1): 434–440.

Amaral, A. R. S. 2006. "On Exact Solution of Facility Layout Problem." *European Journal of Operational Research* 173: 508–518. doi:10.1016/j.ejor.2004.12.021.

Amaral, A. R. S. 2009. "A New Lower Bound for Single Layout Problem." *Discrete Applied Mathematics* 157: 183–190. doi:10.1016/j.dam.2008.06.002.

Amaral, A. R. S. 2012. "Corridor Allocation Problem." *Computers and Operations Research* 39: 3325–3330. doi:10.1016/j.cor.2012.04.016.

Amaral, A. R. S. 2013. "Optimal Solutions for the Double-Row Layout Problem." *Optimization Letters* 7 (2): 407–413. doi:10.1007/s11590-011-0426-8.

Anjos, M. F., A. Kinning, and A. Vannelli. 2005. "A Semidefinite Approach for the Single Row-Layout Problems with Unequal Dimensions." *Discrete Applied Mathematics* 2: 113–122.

Anjos, M. F., and A. Vannelli. 2008. "Computing Globally Optimal Solutions for Single-Row Layout Problems Using Semidefinite Programming and Cutting Planes." *NFORMS Journal on Computing* 20 (4): 611–617. doi:10.1287/ijoc.1080.0270.

Chen, J. C., Y. Y. Chen, T. L. Chen, and J. Z. Lin. 2016. "Comparison of Simulated Annealing and Tabu-Search Algorithms in Advanced Planning and Scheduling Systems for TFT-LCD Colour Filter Fabs." *International Journal Computer Integrated Manufacturing* 1–19. doi:10.1080/0951192X.2016.1145805.

Chung, J., and J. M. A. Tanchoco. 2010. “The Double Row Layout Problem.” *International Journal Products Researcher* 48 (3): 709–727. doi:10.1080/00207540802192126.

Das, S. 1993. “A Facility Layout Method for Flexible Manufacturing Systems.” *International Journal Pro Researcher* 31: 279–297. doi:10.1080/00207549308956725.

Datta, D., A. R. S. Amaral, and J. R. Figuiera. 2011. “Single Row Facility Layout Problem Using a Permutation-Based Genetic Algorithm.” *European Journal of Operational Research* 213 (2): 388–394. doi:10.1016/j.ejor.2011.03.034.

Dorigo, M., V. Maniezzo, and A. Colorni. 1996. “The Ant System: Optimization by a Colony of Cooperating Agents.” *IEEE Transactions on Systems, Man. and Cybernetic-Part B* 26 (1): 29–41. doi:10.1109/3477.484436.

Ficko, M., M. Brezocnik, and J. Balic. 2004. “Designing the Layout of Single and Multiple-Rows Layout Flexible Manufacturing System by Genetic Algorithm.” *Journal of Materials Processing Technology* 157: 150–158. doi:10.1016/j.jmatprotec.2004.09.012.

Gen, M., K. Ida, and C. Cheng. 1995. “Multi-Row Machine Layout Problem in Fuzzy Environment Using Genetic Algorithms.” *Comparative and Industrial Engineering* 29: 519–523. doi:10.1016/0360-8352(95)00127-M.

Heng, L., and P. E. D. Love. 2000. “Genetic Search for Solving Construction Site-Level Unequal-Facility Layout Problems.” *Automation in Construction* 9: 217–251. doi:10.1016/S0926-5805(99)00006-0.

Heragu, S. S., and A. S. Alfa. 1992. “Experimental Analysis of Simulated Annealing Based -Algorithm for Facility Layout Problem.” *European Journal of Operational Research* 57: 1190–1202. doi:10.1016/0377-2217(92)90042-8.

Kirkpatrick, S., C. D. Gelatt Jr., and M. P. Vecchi. 1983. “Optimization by Simulated Annealing.” *Science* 220: 671–680. doi:10.1126/science.220.4598.671.

Kumar, S., P. Asokan, S. Kumanan, and B. Varma. 2008. “Scatter Search Algorithm for the Single Row Facility Layout Problem in FMS.” *Advances in Production Engineering and Management* 3 (4): 193–204.

Murray, C. C., A. E. Smith, and Z. Q. Zhang. 2013. “An Efficient Local Search Heuristic for the Double Row Layout Problem with Asymmetric Material Flow.” *International Journal of Production Research* 51 (20): 6129–6139. doi:10.1080/00207543.2013.803168.

Samarghandi, H., and K. Eshghi. 2010. “An Efficient Tabu Algorithm for the Single-Row Facility Layout Problem.” *European Journal of Operational Research* 205: 98–105. doi:10.1016/j.ejor.2009.11.034.

Samarghandi, H., P. Taabayan, and F. F. Jahantigh. 2010. “A Particle Swarm Algorithm Optimization for the Single-Row Facility Layout Problem.” *Computers and Industrial Engineering* 58: 529–534. doi:10.1016/j.cie.2009.11.015.

Solimanpur, M., B. Vrat, and R. Shankar. 2005. “An Ant Algorithm for the Single Row Layout Problem in Flexible Manufacturing Systems.” *Computers and Operations Research* 32: 583–598. doi:10.1016/j.cor.2003.08.005.

Tompkins, J. A., J. A. White, A. B. Yavuz, H. F. Edward, J. M. A. Tanchoco, and J. Trevino. 1996. *Facility Planning*. 2nd ed. New York: John *Wiley & Sons Inc*.

Tubaileh, A. S. 2014. “Layout of Flexible Manufacturing Systems Based on Kinematic Constraints of the Autonomous Material Handling System.” *International Journal Advancement Manual Technical* 74 (9): 1521–1537. doi:10.1007/s00170-014-6063-5.

Yossef, H., S. Sait, and H. Adiche. 2001. “Evolutionary Algorithm, Simulated Annealing and Tabu Search: A Comparative Study.” *Engineering Applications of Artificial Intelligence* 14: 167–181. doi:10.1016/S0952-1976(00)00065-8.

Zuo, X. Q., C. C. Murray, and A. E. Smith. 2014. “Solving an Extended Double Row Layout Problem Using Multi-Objective Tabu Search and Linear Programming.” *IEEE Transactions on Automation Science and Engineering* 99: 1–11.