PALADYN Journal of Behavioral Robotics

VERSITA

Research Article · DOI: 10.2478/s13230-012-0008-1 · JBR · 2(3) · 2011 · 164-174

Genetic algorithm using a modified backward pass heuristic for the dynamic facility layout problem*

Kazi Shah Nawaz Ripon†, Kyrre Glette, Dirk Koch, Mats Hovin, Jim Torresen

Department of Informatics, University of Oslo, Oslo, Norway

Received 2011/11/08

Accepted 2012/01/20

Abstract

Layout planning in a manufacturing company is an important economical consideration. In the past, research examining the facility layout problem (FLP) generally concerned static cases, where the material flows between facilities in the layout have been assumed to be invariant over time. However, in today's real-world scenario, manufacturing system must operate in a dynamic and market-driven environment in which production rates and product mixes are continuously adapting. The dynamic facility layout problem (DFLP) addresses situations in which the flow among various facilities changes over time. Recently, there is an increasing trend towards implementation of industrial robot as a material handling device among the facilities. Reducing the robot energy usage for transporting materials among the facilities of an optimal layout for completing a product will result in an increased life for the robots and thus enhance the productivity of the manufacturing system. In this paper, we present a hybrid genetic algorithm incorporating jumping genes operations and a modified backward pass pair-wise exchange heuristic to determine its effectiveness in optimizing material handling cost while solving the DFLP. A computational study is performed with several existing heuristic algorithms. The experimental results show that the proposed algorithm is effective in dealing with the DFLP.

Keywords

dynamic facility layout problem (DFLP) · modified backward pass pair-wise exchange heuristic · jumping genes operations · material handling (MH) cost · industrial robot

1. Introduction

The facility layout problem (FLP) is the determination of the most efficient physical arrangement of a number of facilities within a manufacturing plant to minimize a given objective while satisfying some constraints. A typical manufacturing plant has a number of facilities interacting with each other. A facility is an entity that assists in one dedicated task [1]. For example, such a facility may be a department, a machine tool, a work center, a manufacturing cell, a machine shop, or a warehouse. The FLP is a combinatorial problem of high complexity and the runtime for solving it quickly increases with the number of facilities to be laid out. Layout planning in a manufacturing company is not only a computationally difficult task, but also has a significant economical contribution towards the manufacturing system.

Huge deployment of robots in manufacturing industries is very common these days [2]. 90% of all robots used today are found in factories and they are referred to as industrial robots [3]. This is mainly for the economic benefits by reducing operating cost through material handling [4]. In addition, other advantages of robots in manufacturing systems include improving product quality and consistency, coping with complex production system, increasing production output rates and manufacturing flexibility, reducing material waste, and minimizing human involvement in the production system. In fact, the overall performance of the robots in a manufacturing plant mainly depends on the optimal arrangement of facilities on the manufacturing layout. Adapting the FLP to robotics, an effective facility layout design reduces manufacturing lead time for the robots and increases the throughput, hence increases overall productivity and efficiency of the plant.

To stay alive in business under global market competition, every company strives to reduce its costs. Material handling (MH) cost constitutes the major part of total operating costs. For a given layout, MH costs are determined based on the sum of the product of materials flow, distance, and transportation cost per unit per distance unit for each pair of facilities. It has been estimated that MH costs are between 20% and 50% of the total operating cost and between 10% and 80% of the total cost of manufacturing a product [5]. Therefore, even small improvements in MH costs can create a major impact on lowering the total operating costs. Thus, the most common objective considered in FLPs is the minimization of MH costs. Therefore, to reduce product costs and improve the effectiveness of the manufacturing process, it is essential to have an optimal arrangement of facilities.

Over the years, extensive research has been conducted on the FLP. Unfortunately, most of the research conducted on the FLP is typically focused on the static FLP (SFLP), where the material flows between facilities in the layout have been assumed to be fixed over time [6, 7]. However, most manufacturing facilities today operate in a dynamic and market-driven environment in which production rates and production mixes are continuously adapted. Layouts are constantly changing, either in response to customers' demands for changes in product designs and functionalities or to keep pace with technological innovations. The introduction of new products/machines and the removal of others, as well as the realization of an increase or decrease in throughput volume can render the existing layout completely unreliable in yielding improved productivity. MH costs are no longer constant

*The paper has been partially published in IEEE 9th International Conference on Cybernetic Intelligent Systems (CIS), 2010, Reading, UK.

†E-mail: ksripon@ifi.uio.no

over the planning horizon, creating the need for radical modifications in layouts. It is therefore often necessary to analyze and redesign the current layout to accommodate the constant changes in flow and demand. The dynamic FLP (DFLP) extends the SFLP by taking into account the changes in material flow over the planning horizon and the cost of rearranging the layout.

A material handling system ensures the delivery of material to the appropriate facilities. The selection of the material handling equipment is important in the design of a modern manufacturing system and can include robots, automated guided vehicles, conveyor systems, or others [8]. Recently, there is an increasing trend towards implementation of industrial robot as a material handling device for manufacturing plants [9]. In a typical manufacturing process, a robot is used for transporting materials between facilities involved in the repetitive sequence of operations required to produce manufacturing product. The fact that the productivity of a manufacturing system is dependent on the performance of industrial robots is widely known [10]. In the manufacturing layout where robots are used, it is possible to save the energy used by the robots if the facilities are placed in an optimized way. A great advantage of industrial robots is their versatility, and robots can easily cope up with the constantly changing nature of the modern DFLP. This is because; industrial robots have a high flexibility to adapt to the large fluctuations of transportation sequence and an ability to carry out various amounts of materials [11]. As mentioned earlier, the MH cost is a function of the distance the material is moved between facilities in a manufacturing plant. To reduce this cost, it is essential to have an optimal arrangement of facilities which can minimize the total distance travelled. Similarly, the necessities of having an optimal layout is of the same importance for the industrial robot where the material flows between facilities affect the sequence of transportation made by the robot between facilities [12]. Consequently, minimizing the MH cost (i) enhances the production rate and profit of the manufacturing system, (ii) saves the robot energy usage, and (iii) increases the life of the robot.

Problems related to facility layout are computationally difficult. It is one of the classic computer science problems and has been shown to be NP–hard [13, 14]. In an n–facility, t–period DFLP, the maximum number of different layouts is $(n!)^t$. We would have to evaluate 1.93×10^{14} possibilities for even just a 6–facility, 5–period DFLP. The inherent difficulty of the DFLP is in the large number of combinations of facilities for every period of the planning horizon that is representative of practical production systems. Due to the combinatorial nature of the problem, exact approaches have been successfully applied only to small problems, but they require high computational efforts and extensive memory requirements. As a result, many researchers are led to near-optimal heuristics and meta-heuristics for searching through the huge search space (like large scale DFLPs). Interested readers should consult [13, 15] for a detailed review. Among those approaches, the genetic algorithm (GA) has found a wide application in research intended to solve the DFLP due to its capability to generate feasible solutions in a minimum amount of time [13].

The GA [16] is a robust search and optimization technique that starts with a population of randomly generated candidate solutions and uses probabilistic rules to evolve a population from one generation to the next. GAs have been known to offer significant advantages against conventional methods in developing near-optimal solutions by using simultaneously and inherently parallel search principles and heuristics. GAs have been widely applied for optimization in many fields, including engineering, physical sciences, social sciences, and operations research. Consequently, GAs have been successful in obtaining near-optimal solutions to many different combinatorial optimization problems. Generally speaking, the GA outperforms other heuristic and meta-heuristic methods due to its capability to generate feasible solutions in a minimum amount of time, and seems to have become quite popular in solving both SFLPs and DFLPs [13, 17].

In this paper, we propose a GA for solving the DFLP using a modified version of backward pass pair-wise exchange heuristic [18]. In fact, this approach extends our previous approach to the DFLP [7], which was a hybrid GA based approach incorporating jumping gene operations [19, 20]. The proposed heuristic is based on Urban's steepest descent pair-wise exchange heuristic for the DFLP [21]. Experimental results show that Urban's method provides good solutions. However, it is only forward pass in nature. Thus, the quality of the layouts for later periods is completely dependent on the quality of their preceding layouts. The introduction of backward pass approach can be an option to remove this limitation. We implemented a modified backward pass pair-wise exchange heuristic that fits into framework of the DFLP and the GA. The central idea for incorporating jumping genes operations is to fine-tune solutions during evolution in the form of a local search. It is particularly effective for long chromosomes that are customary in large DFLPs.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature for the DFLP. Section 3 justifies the importance of the modified version of backward pass pair-wise exchange heuristic for the DFLP, as well as the implementation of this heuristic. Section 4 offers a brief overview of the jumping genes operations and their effects on the DFLP. Section 5 presents the mathematical formulation for the DFLP adapting to robot as a material handling device. Section 6 outlines the implementation of our proposed DFLP approach. Section 7 presents and analyzes the experimental results. Finally, this paper ends with conclusions in Section 8.

2. Literature review

Rosenblatt [22] was the first to address the basic DFLP and sketched out a method based on the dynamic programming (DP) to solve the DFLP. However, this approach is not practical for large problems. Since then, researchers have been developing a large number of the DFLP approaches, considering both exact and heuristic approaches, recognizing their importance. Comprehensive surveys are found in [13, 23, 24]. Urban [21] proposed a steepest-descent, pair-wise exchange heuristic for the DFLP. Modifications of this heuristic have been carried out by Balakrishnan et al. [18]. Balakrishnan et al. [25] proposed a network programming model by adding the constraint of a budget for the total rearrangement costs over the entire finite horizon. Given that exact approaches are computationally intractable while solving large FLPs, most existing solution approaches are based on heuristic due to their ability to generate feasible solution in the least possible computational time. Since Conway and Venkataramanan [26] first examined the suitability of the GA for the DFLP, it has become very popular among researchers. Though their algorithm was applied only to 6 and 9 facilities, it performed better than the DP. Balakrishnan and Cheng [27] refined the GA approach presented by Conway and Venkataramanan [26] employing a nested loop GA. Balakrishnan et al. [28] further extended the existing GA applications for the DFLP to a hybrid GA. A recent application of the GA using new crossover and mutation operators can be found in [29]. Kaku and Mazzola [30] applied the tabu search heuristic to the DFLP in a two-stage search process. Erel et al. [31], Baykasoglu and Gindy [32], and McKendall et al. [33] applied the simulated annealing (SA) for the DFLP, and obtained encouraging results. More recently, McKendall and Shang [6] applied a hybrid ant colony algorithm and Rezazadeh et al. [34] applied an extended particle swarm algorithm for the DFLP. In this paper, we compare our proposed algorithm with the recent approaches so far available in the literature and get very promising results.

3. Modified backward pass pair-wise exchange heuristic

The heuristic used in this work is influenced by Urban [21]. Urban's heuristic is a steepest descent pair-wise exchange heuristic incorporating the principle for forecast windows for solving the DFLP. The forecast window is the number of periods being considered when the pair-wise exchange is performed. The length of the forecast windows (m) varies between 1 and the number periods in the planning horizon. This procedure combines the SFLP and the DFLP into one. An initial layout is given only for the first period of the planning horizon. Using this layout and pair-wise exchanges, one set of multi-period layouts is obtained for the given planning horizon in each forecast window. The flow data for one or more periods are combined and used to determine a layout for the current period. For instance, when m = 1, the flow data for period 1 is used to improve the given initial layout by pair-wise exchanges. Then, pair-wise exchanges are used again to determine a "good" layout for period 2 using the newly generated improved layout for period 1 along with the flow data for period 2. After obtaining a layout for each period in the planning horizon for m = 1, m is set to 2. Similarly, for m = 2, the flow data for time periods 1 and 2 are combined to find an improved layout for time period 1 using pair-wise exchanges. The flow data for time periods 2 and 3 are used to obtain a layout for time period 2 and so on. The length of the forecast window is incrementally extended until it equals the entire planning horizon, and the layout of the last period is improved. Therefore, there will be m layouts for m forecast windows and the solution with the minimum MH cost is selected. This heuristic decreases the computational complexity and usually obtains good solutions even for large size problems. Interested readers are referred to [18, 21] for the details.

Algorithm 1 Modified backward pass pair-wise exchange heuristic

```
set F_1 = total MH cost for the current chromosome
for i = P down to 1 do
  f_1 = MH cost for the current period
  for j = 1 to n − 1 do
    k = j + 1
    construct a new layout for period i by swapping j and k
    f_2 = MH cost of the changed layout for this period
    if f_2 < f_1 (considering rearrangement cost also) then
      store the new layout for this period in D
    end if
  end for
  if there is any element in D then
    find the exchange that produces the lowest MH cost for this period
    store the layout for this period in D̄
  end if
end for
if there is any element in D̄ then
  calculate F_2 = total MH cost for the changed chromosome
  if F_2 < F_1 then
    replace the current chromosome with it
  end if
end if
```

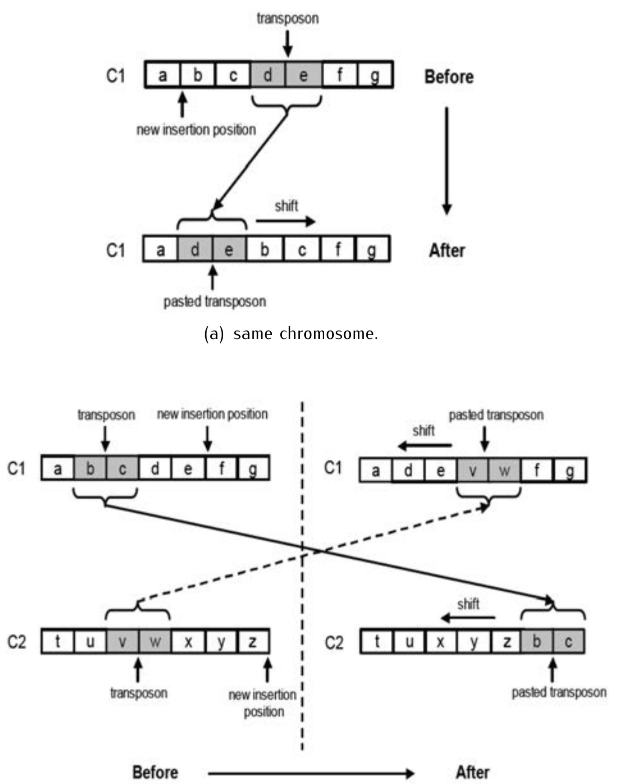
However, Urban's heuristic is naturally a forward pass heuristic. Once a layout for a period is determined, it is never changed in subsequent periods. Thus, the quality of the layouts in the later periods, and consequently the final solution, depends on the preceding layouts greatly. An improvement that works backward, starting from the final solutions, to search for better solutions was proposed by Balakrishnan et al. [18, 23]. In this technique, Urban's heuristic is used to solve the DFLP for each m. Then a backward pass pair-wise exchange is performed on each of these m solutions. The best of these solutions is selected as the final layout. In this work, we employed a new backward pass approach with modified pair-wise exchange for solving the DFLP. Unlike Urban [21] and the improved version proposed in [18], we employed pair-wise exchanges only in the backward pass. This is because, we obtain our solutions in the forward pass using the GA. And, according to the GA theory, the final solutions obtained by the GA (forward pass, in this case) should be optimal after completing the evolutionary cycle [35]. Existing results also justify this claim [13, 17]. Therefore, employing the pair-wise exchanges for the forward pass is nothing but mere waste of computational time and effort.

Since the final layout is almost optimal, we set the length of forecast window (m) equal to 1 in the backward pass (similar to [23]). That means, material flows from different periods are never added in performing pair-wise exchanges. In a traditional pair-wise exchange, a large number of comparisons are also required. An n–facility layout has ${}_nC_2$ possible pairs. And for large DFLPs with long planning horizons, this number will be huge. Instead of comparing every gene of a chromosome, we also employed a new pair-wise exchange procedure. We only exchange the genes with a common boundary, if the exchange provides a lower MH cost for the layout (considering rearrangement cost also). The idea behind the modified pair-wise exchange is that the layouts performing in the backward pass is already almost optimal. Therefore, the backward pass will never generate a layout that produces a higher MH cost than the current cost. If MH cost can not be reduced, no pair-wise exchange will take place. As discussed earlier, certainly it will save the computational time and effort. The general outline of our proposed pair-wise exchange heuristic for the DFLP is given in Algorithm 1. Here, P and n denote the number of periods in the planning horizon and the total number of facilities, respectively.

4. Jumping genes genetic algorithm (JGGA)

Mimicking the jumping genes operations in biological chromosomes discovered by Nobel Laureate Barbara McClintock, the jumping genes genetic algorithm (JGGA) is a novel evolutionary algorithm (EA) that has recently been proposed [36]. The most important feature of the JGGA is its capability to exploit local search heuristics by emulating a genetic phenomenon of horizontal transmission in which genes can jump from one position to another either within its own or to the other chromosomes. Every conventional genetic operator in the GA employs only vertical transmission of genes from generation to generation. However, the jumping genes operators introduce a kind of horizontal transmission. Indeed, the jumping genes operations are better ways for exploration and exploitation than the usual genetic operators only. Therefore, these operations create more chances to achieve better convergence and diversity, as well as to avoid premature convergence [19, 20]. Recently, many successful applications of the JGGA have been reported in the literature [37–40].

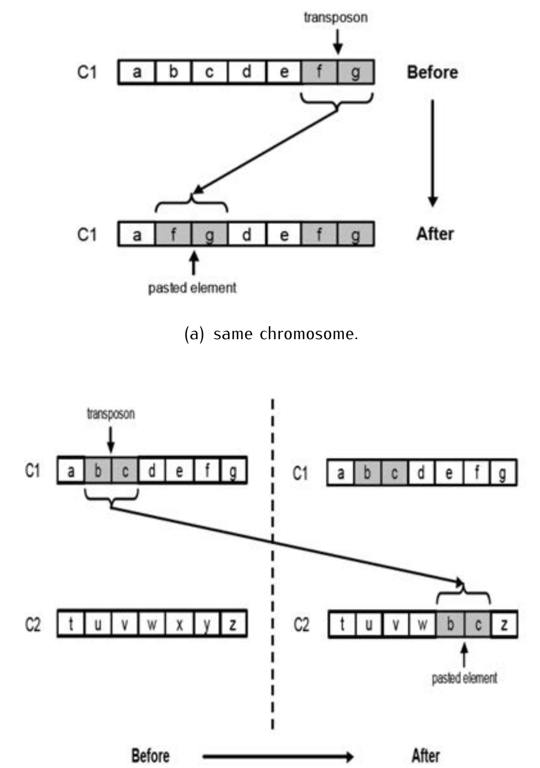
Its success is mainly due to the two newly designed computational operations — the copy and paste & the cut and paste to emulate the jumping behavior (transposition process) into an EA framework. To implement these operations, some consecutive genes are selected as a transposon. The actual implementation of the cut and paste operation



(a) same chromosome.

(b) different chromosome.

Figure 1. Cut and paste transposition.



(a) same chromosome.

(b) different chromosome.

Figure 2. Copy and paste transposition.

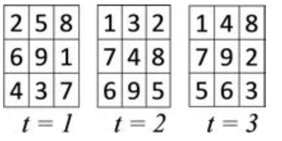


Figure 3. A 3×3 DFLP instance with 9–facility, 3–period.

is that the transposon is cut from the original site and pasted into a new site (Fig. 1). In the case of the copy and paste operation, the transposon replicates itself, with one copy of it inserted into a new site, while the original one remains unchanged at the same site (Fig. 2). For the detailed description of the JGGA, the reader may refer to [41]. It is well known that GAs are not very effective for fine-turning the solutions that are already close to the optimal solution as crossover and mutation may not be sufficient enough to generate feasible solutions. Hence, it is necessary to integrate some local search strategies in the GA for enhancing the solutions. In addition, it should be noted that as the length of a chromosome increases with the size of the DFLP, GAs might suffer from premature convergence in a large search space. To tackle this, the jumping genes operations offer a local search capability to exploit solutions around the chromosomes, while the usual genetic operators globally explore solutions from the population.

5. Problem formulation

Traditionally the FLP has been presented as a quadratic assignment problem (QAP) [17, 42]. In this formulation, n equal area facilities are assigned to n locations with the constraint that each facility is restricted to one location and one facility should choose only one location. The objective of the DFLP is to obtain layouts for each period in the planning horizon such that the sum of the rearrangement and MH costs is minimized. Consequently, the DFLP can be formulated as an extension of the SFLP by selecting a static layout for each period and then deciding whether to change to a different layout in the next period or not. If the rearrangement cost were negligible, the optimal solution would have been to combine the optimal static layout for each period of the planning horizon. However, in reality, layout rearrangement incurs costs. Hence, the DFLP is not just a series of the SFLP. Since the total cost will be calculated based on the entire planning horizon, the layout for each period influences the layouts for the other periods. So, the DFLP involves selecting a static layout for each period and then deciding whether to change to a different layout in the next period. Fig. 3 presents a 3×3 DFLP instance with 9–facility and 3–period. In this example, facilities $2, 5, 8, 6, 9, 1, 4, 3,$ and 7 are assigned to locations $1, 2, 3, 4, 5, 6, 7, 8,$ and 9 respectively in period 1, and so on.

Solving a complete layout problem with all the details in an efficient way is quite unlikely. Therefore, it is quite common that researchers make several assumptions and simplifications in their models without missing the important underlying structure [43]. In this work, we follow the assumptions described in [28]: facilities and locations are equal-sized; the number of periods in the planning horizon is known; and the distances between the facilities are determined a priori. As stated earlier, the DFLP tries to minimize the total MH cost along with the rearrangement costs. Adapting the DFLP to robotics again, the target is to find the optimal layouts for the total planning horizon. Hence, the measure to be minimized is the transportation path which ensures that the robots pass through the facilities following the orders for different periods with the minimum possible usage of energy. Since robots are capable of lifting hundreds of pounds of payload and positioning the weight with accuracy to a fraction of a millimeter [44], this will also save the material usage, and increase the product quality and throughput. A mathematical formulation for the DFLP is shown below:

$$F_1 = \sum_{t=2}^{P}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} A_{tijl}Y_{tijl} + \sum_{t=1}^{P}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{tik}d_{tjl}X_{tij}X_{tkl}E_{tik} \quad (1)$$

Subject to

$$\sum_{j=1}^{n} X_{tij} = 1;\ i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, P; \quad (2)$$

$$\sum_{i=1}^{n} X_{tij} = 1;\ j = 1, 2, \ldots, n;\ t = 1, 2, \ldots, P; \quad (3)$$

$$Y_{tijl} = X_{(t-1)ij}X_{til};\ i, j, l = 1, 2, \ldots, n;\ t = 2, 3, \ldots, P; \quad (4)$$

$$X_{tij} = \begin{cases} 1; & \text{if facility } i \text{ is at location } j \text{ in period } t \\ 0; & \text{otherwise} \end{cases} \quad (5)$$

$$Y_{tijl} = \begin{cases} 1; & \text{if facility } i \text{ is shifted from location } j \text{ to } l \text{ in period } t \\ 0; & \text{otherwise} \end{cases} \quad (6)$$

Where, i, k are facilities; j, l are locations in the layout; A_{tijl} is the fixed cost for shifting facility i from location j to l in period t (where $A_{tijj} = 0$); f_{tik} is the flow cost for unit distance from facility i to k in period t; d_{tjl} is the distance from location j to l in period t; E_{tik} is the energy used by the robots for transporting unit material from facility i to k in period t; n is the total number of facilities in the layout; and P is the number of periods for the planning horizon.

It should be noted that, this objective includes the energy used by the robots for completing a complete transportation sequence for every period of the planning sequence. However, for the lack of benchmark problem or real-world data, and for comparing with existing methods, we fixed E_{tik} = 1. Thus, the objective is to minimize the sum of the layout rearrangement cost (first term) and the MH costs (second term) over the planning horizon (Eq. 1). Constraints Eqs. 2 and 3 ensure that each location is assigned to exactly one facility and each facility is assigned to only one location at each period, respectively. Constraint Eq. 4 assigns a value 1 only if a facility is shifted between locations in consecutive periods.

6. The proposed approach

6.1. Chromosome representation

The initial population is generated randomly, and we consider a form of direct representation for chromosomes. The solution is represented as a string of integers of length $n \times P$, where n is the total number of facilities, and P is the number of periods for the planning horizon. The chromosomes are encoded as $(a_{11}a_{12}a_{13}, \ldots, a_{1n})$ $(a_{21}a_{22}a_{23}, \ldots, a_{2n})$,. . . . . ., $(a_{P1}a_{P2}a_{P3}, \ldots, a_{Pn})$. The integers denote the facilities and their positions in the string denote the positions of the facilities in the layout for that period. This representation is very useful for the GA because a chromosome will be chosen for crossover, mutation or jumping operations if it has a good objective value and also the objective value for the chromosome is easy to calculate. Fig. 4 illustrates the chromosome for the 9–facility, 3–period DFLP mentioned in Fig. 3.

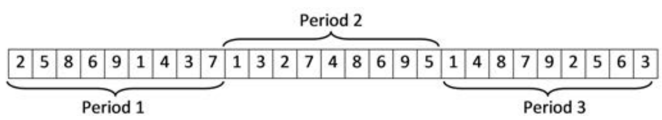


Figure 4. Chromosome representation for the 9–facility, 3–period DFLP presented in Fig. 3.

6.2. Crossover and mutation

In this approach, we applied a single point crossover operation. For keeping the chromosomes valid after the operation, some repair operations are required to remove any duplication or absence of facilities. These repair operations are only required if the crossover point is within the genes of a particular period. However, no repair operation is necessary when the crossover point is the first or the last gene of any period (the boundary of two periods, in other words). In the repairing process, first we find and list the duplicate facilities in that particular period according to the positions in the chromosome. Then, we check whether any facilities are missing in that period (starting from the first to the last gene of the period). After that, we replace the list of the duplicate facilities with facilities that are missing. For mutation, we use the traditional swap mutation with the restriction that two genes are chosen randomly from the same time period. As a result, the resulting chromosome is legal and no further repair is required.

6.3. Jumping genes operations

In order to apply the jumping genes operations for solving the DFLP, we consider the total number of genes for a period as a single transposon. The length of each transposon can be more than one period and the transposition operations are chosen randomly. Furthermore, the transpositions made within the same or to a different chromosome are also selected randomly. To apply the cut and paste operation into a pair of chromosomes, a transposon from each chromosome is selected randomly. Then, for both chromosomes, the transposon is cut from its original position and pasted into the new position of the other chromosome. The inserting positions are selected randomly for each chromosome with the restriction that it must be the starting gene of any period. The gaps created because of cutting the transposon is replaced by shifting the neighbor genes as depicted in Fig. 1(b).

This operation can also be carried out for a single chromosome in a similar way. In the case of the copy and paste, the transposon is randomly selected from any of the two chromosomes. A replication of the gene pattern is then inserted into a new location of the other chromosome, while the original one remains unchanged at the same site as depicted in Fig. 2(b). The new location is selected randomly with the restriction mentioned above. Similarly, the operation on the same chromosome is depicted in Fig. 2(a). It should be noted that, the jumping genes operations are chosen randomly on the basis of probability like other genetic operations. Also, the type of operations and the transpositions made within the same chromosome or to a different chromosome are chosen randomly and there is no restriction to the chromosome choice.

7. Computational results

This section presents the computational results of the proposed GA based approach with a modified version of backward pass pair-wise exchange heuristic. As discussed in Section 5, there is no benchmark dataset or published result for comparing the performance of the DFLP, even the SFLP, adapted to robotics. Therefore, we could not compare the performance of our proposed approach for this framework. However, by setting the parameter $E_{tik} = 1$ in Eq. 1, we compare our approach with the 48 test problems obtained from Balakrishnan and Cheng [27]. These problems were originally designed for the DFLP and the objective is to minimize the total MH cost. In fact, this comparison is reasonably fair in the sense that it will correct the objective function achieved by every comparing methods by the same amount based on the energy used by the robots for transporting unit material from facility i to k in period t. The test problems are divided into six types with 6, 15, and 30 facilities with 5 and 10 time periods for each. Each type has eight instances. This will give us an idea of the effectiveness of the algorithm for varying sized layouts and over varying sized planning horizons.

The experiments are conducted using 200 chromosomes and 400 generations for DFLPs with up to 15 facilities, and 400 chromosomes and 1000 generations for DFLPs with more than 15 facilities. The probabilities of crossover, mutation and jumping operations are 0.8, 0.2, and 0.6, respectively. We use the traditional tournament selection with the tournament size of 2. Each problem is tested for 30 times with different seeds and the best and average solutions are recorded. To justify the efficiency of the proposed approach, the results are compared to both GA–based and other evolutionary and heuristic DFLP approaches. Tables 1∼6 summarize the results of the comparisons. These tables are partially cited from [7]. The results are compared to the results obtained by our previous hybrid GA approach (LHGA) [7], the GA presented by Conway and Venkataramanan (CVGA) [26], the nested-loop GA (NLGA) [27], the hybrid GA presented by Balakrishnan et al. (GADP) [28], the SA heuristic (SA) [32], the DP presented by Erel et al. (DP) [31], the hybrid ant systems (HAS) [6], and the discrete particle swarm optimization based heuristic (DPSO) [34]. The bold numbers give the best solution for each test problem. Since, we don't have the average values for comparing algorithms except the proposed approach and LHGA; we are unable to mention their average values. However, superior performances of LHGA have been already demonstrated in our previous paper [7]. Therefore, for average values, we only compare our proposed approach with LHGA.

In the 6–facility DFLPs, the results for 5–period and 10–period problems are shown in Tables 1 and 2, respectively. From the tables, we can find that our proposed approach obtains the best solutions for all the 16 problems of both categories (Problems 1∼16). For the 5–period problems (Problems 1∼8), LHGA, DPSO and HAS also find the best results for all 8 problems. However, for the 10–period problems (Problems 9∼16), LHGA, GADP, DPSO, HAS and DP obtain the best solutions for 4, 3, 5, 5, and 5 of the 8 problems, respectively. In fact, our proposed approach finds new best solutions for 2 problems (Problems 13 & 16). The average values are also better than or are in par with LHGA.

Tables 3 and 4 report the results for 15–facility with 5 and 10 period problems, respectively. For the 5–period problems (Problems 17∼24), the proposed approach and DPSO both obtain the best solutions for 5 out of the 8 problems. Where as, LHGA and HAS obtain the best solution(s) for 2 and 1 of the 8 problems. All other algorithms fail to achieve any best solution. Interestingly, the results obtained by DPSO are not so good for the 10–period DFLPs (Problems 25∼32). Here, DPSO fails to achieve any best solution. However, the performance of our proposed approach is consistent. Clearly, our proposed approach and SA outperform other approaches for the 10–period problems. Both approaches obtain the best solutions for 4 of the 8 problems. Our previously proposed LHGA finds 2 best solutions. Our proposed approach with the modified heuristic also performs the same for these 2 problems (Problems 27 & 31). In the 16 problems of these combinations, the proposed approach obtains the new lowest MH cost for 4 problems. From the Tables, we can find that the problems where our proposed approach fails to achieve the best results, its performances are not too inferior. In fact, the deviation is only 1% in the worst case. Considering both 5 and 10 period problems, the performance of our algorithm is relatively stable. If we compare our proposed approach only with the previously proposed LHGA, the results are significantly better than the results obtained by LHGA. Furthermore, the proposed approach improves the objectives in comparison to LHGA for most of the problems, and never performs worse than LHGA. As usually, the average values obtained by our proposed approach for both 5 and 10 period problems are significantly better.

The results for 30–facility with 5 and 10 period DFLPs are shown in Tables 5 and 6, respectively. For the 5–period problems (Problems 33∼40), the proposed approach obtains the best solutions for 5 of the 8 problems. While, LHGA, SA, DPSO, HAS obtain the best solution(s) for 3, 1, 1, and 1 of the 8 problems. Clearly, the proposed approach outperforms all other algorithms. On the other hand, for the 10–period problems (Problems 41∼48), both the proposed approach and SA perform quite well. Each obtains 4 of the 8 best solutions for this combination. Whereas, other approaches fail to obtain any of the best solutions. Then again, if we compare the proposed approach against LHGA, we can find that the proposed approach significantly outperforms LHGA. In fact, the proposed approach not only finds 4 new best solutions, but also improves the objectives for almost every case of the 16 problems of this combination while comparing with the results of LHGA. Indeed, this is the overall trend for all the combinations of test problems. Simultaneously, the average values obtained by the proposed approach are significantly better than the average values obtained by LHGA.

It is natural to raise the concern about the additional coding effort, and consequently the additional complexity for the proposed modified heuristic within the framework of LHGA [7]. For justifying the optimization behavior of the proposed approach, again we run the proposed approach with the modified version of heuristic for less than the mentioned generations (70% of the scheduled generations for all test problems). Since LHGA has been already justified as a superior approach in comparison to other comparing approaches [7], here we compare our proposed approach only with LHGA by running the same reduced evolutionary cycles for LHGA. Fig. 5 presents these

Table 1. Total cost of 6–facility and 5-period problems

Pr.	CVGA	NLGA	GADP	SA	DPSO	HAS	DP	LHGA		Proposed		
								Best	Avg	Best	Avg	Deviation
1	108976	**106419**	**106419**	107249	**106419**	**106419**	**106419**	**106419**	106419	**106419**	106419	0 %
2	105170	**104834**	**104834**	105170	**104834**	**104834**	**104834**	**104834**	104834	**104834**	104834	0 %
3	104520	**104320**	104529	104800	**104320**	**104320**	**104320**	**104320**	104320	**104320**	104320	0 %
4	106719	106515	106583	106515	**106399**	**106399**	106509	**106399**	106399	**106399**	106399	0 %
5	**105628**	**105628**	**105628**	106282	**105628**	**105628**	**105628**	**105628**	105628	**105628**	105628	0 %
6	105606	104053	104315	**103985**	**103958**	**103958**	**103958**	**103985**	103985	**103985**	103985	0 %
7	**106439**	106978	106447	106447	**106439**	**106439**	106447	**106439**	106439	**106439**	106439	0 %
8	104485	**103771**	**103771**	**103771**	**103771**	**103771**	**103771**	**103771**	103771	**103771**	103771	0 %

Table 2. Total cost of 6–facility and 10-period problems

Pr.	CVGA	NLGA	GADP	SA	DPSO	HAS	DP	LHGA		Proposed		
								Best	Avg	Best	Avg	Deviation
9	218407	214397	**214313**	215200	**214313**	**214313**	**214313**	**214313**	214313	**214313**	214313	0 %
10	215623	212138	**212134**	214713	**212134**	**212134**	**212134**	**212134**	212134	**212134**	212134	0 %
11	211028	208453	**207987**	208351	**207987**	**207987**	**207987**	**207987**	207987	**207987**	207987	0 %
12	217493	212953	212741	213331	212530	212530	212741	**212498**	212512	**212498**	212498	0 %
13	215363	211575	210944	213812	210906	210906	211022	205597	205597	**205564**	205564	-0,016 %
14	215564	210801	210000	211213	**209932**	**209932**	**209932**	210364	210412	**209902**	209932	0 %
15	220529	215685	215452	215630	**214252**	**214252**	**214252**	214967	215001	**214252**	214268	0 %
16	216291	214657	212588	214513	212588	212588	212588	205332	205368	**205314**	205344	-0,009 %

Table 3. Total cost of 15–facility and 5-period problems

Pr.	CVGA	NLGA	GADP	SA	DPSO	HAS	DP	LHGA		Proposed		
								Best	Avg	Best	Avg	Deviation
17	504759	511854	484090	484695	**480453**	**480453**	482123	484628	485188	481004	481414	0.114%
18	514718	507694	485352	486141	482568	484761	485702	481940	482210	**481838**	482258	-0,020 %
19	516063	518461	489898	496617	**486658**	488748	491310	489987	490315	**486658**	487108	0 %
20	508532	514242	484625	490869	**480359**	484446	486851	486452	486856	480924	481220	0.117%
21	515599	512834	489885	491501	486658	487722	491178	**486600**	487006	**486600**	486952	0 %
22	509384	513763	488640	491098	**485637**	486685	489847	488894	489184	485694	485986	-0,011 %
23	512508	512722	489378	491350	485462	486853	489155	**484241**	484826	**484241**	484686	0 %
24	514839	521116	500779	496465	**488865**	491013	493577	500516	500962	**488865**	489116	0 %

results. From the figures, we can find that after completing 50% of the scheduled generations, the proposed approach starts finding the best known values for more than half of the populations for all test problems. In fact, it almost convergences for around 70% of the scheduled generations. Where as, at this point, the performances of LHGA are not satisfactory enough. They can find the best values only for small DFLPs and the number of optimal solutions is also small. Thus, the proposed approach appears to be highly effective, and the additional coding effort and complexity required in comparison to LHGA is definitely justified.

To summarize the result, the proposed approach, LHGA, CVGA, NLGA, GADP, SA, DPSO, HAS, and DP find the best solutions for 34, 21, 2, 5, 7, 11, 19, 15, and 11 of the 48 problems, respectively. It appears from the results that LHGA and DPSO are the close competitors in terms of performance. However, the problems for which LHGA finds the best solutions, our proposed approach also finds the same best solutions.

Table 4. *Total cost of* 15–*facility and* 10-*period problems*

Pr.	CVGA	NLGA	GADP	SA	DPSO	HAS	DP	LHGA		Proposed		
								Best	Avg	Best	Avg	Deviation
25	1055536	1047596	987887	**950910**	978546	980351	983070	983537	984370	960306	961014	0.978%
26	1061940	1037580	980638	**947673**	975684	978271	983826	954909	955746	947790	948254	0.012%
27	1073603	1056185	985886	968027	976382	978027	988635	**967608**	968106	**967608**	968026	0 %
28	1060034	1026789	976025	950701	972684	974694	976456	950674	951476	**949062**	949682	-0,169 %
29	1064692	1033591	982778	**948470**	976645	979196	982893	971387	971997	958106	959048	1,005 %
30	1066370	1028606	973912	**948630**	969326	971548	974436	963518	964082	951328	951752	0,283 %
31	1066617	1043823	982872	965844	978657	980752	982790	**965201**	965710	**965201**	965804	0 %
32	1068216	1048853	987789	956170	982964	982707	988584	979381	979908	**955074**	955540	-0,114 %

Table 5. *Total cost of* 30–*facility and* 5-*period problems*

Pr.	CVGA	NLGA	GADP	SA	DPSO	HAS	DP	LHGA		Proposed		
								Best	Avg	Best	Avg	Deviation
33	632737	611794	578689	**562405**	575684	576886	579741	567362	568124	563386	563642	0,174 %
34	647585	611873	572232	569251	570365	570349	570906	**560015**	561338	**560015**	561008	0 %
35	642295	611664	578527	564464	575698	576053	577402	**562528**	562998	**562528**	562780	0 %
36	634626	611766	572057	552684	566124	566777	569596	551936	553396	**551778**	552142	-0,028 %
37	639693	604564	559777	559596	558680	**558353**	561078	568303	568916	561049	561496	0,48 %
38	637620	606010	566792	592515	**565894**	566792	567154	568221	569148	568221	568940	0,409 %
39	640482	607134	567873	582409	567131	567131	568196	**561965**	562956	**561965**	562478	0 %
40	635776	620183	575720	578549	574369	575280	575273	574495	575016	**573674**	573986	-0,121 %

Table 6. *Total cost of* 30–*facility and* 10-*period problems*

Pr.	CVGA	NLGA	GADP	SA	DPSO	HAS	DP	LHGA		Proposed		
								Best	Avg	Best	Avg	Deviation
41	1362513	1228411	1169474	1122154	1161124	1166164	1171178	1122689	1131012	**1121397**	1130124	-0,067 %
42	1379640	1231978	1168878	1120182	1155634	1168878	1169138	**1120170**	1130946	**1120170**	1128680	0 %
43	1365024	1231829	1166366	**1125346**	1158264	1166366	1165525	1163610	1172236	1130416	1141984	0,448 %
44	1367130	1227413	1154192	**1120217**	1144872	1148202	1152684	1121074	1133452	1120704	1131162	0,043 %
45	1356860	1215256	1133561	1158323	1125687	1128855	1128136	1172713	1184864	**1123320**	1139604	-0,21 %
46	1372513	1221356	1145000	1111344	1142568	1141344	1143824	**1110975**	1122954	**1110975**	1121784	0 %
47	1382799	1212273	1145927	**1128744**	1141722	1140773	1142494	1209780	1220124	1148042	1160966	1,68 %
48	1383610	1245423	1168657	**1136157**	1160658	1166157	1167163	1136809	1150532	1136692	1148648	0,047 %

Besides, LHGA never outperforms our proposed approach. In contrast, our proposed approach frequently outperforms LHGA. Considering the performance of DPSO, we can find that in most cases it is better than or equal to our proposed approach for small and medium sized DFLPs. Still, the performance of our proposed approach is satisfactory enough for these problems. Furthermore, for large DFLPs, our proposed approach clearly outperforms DPSO. It can be observed that for large DFLPs, SA performs well; on the contrary, the performances of SA are not satisfactory enough for small and medium sized problems. Quite the opposite results can be observed for HAS and DP. For small DFLPs, they perform well. But, for medium and large DFLPs, their performances are disappointing. Therefore following the overall trend, we can conclude that our proposed hybrid GA incorporating a modified backward pass pair-wise exchange heuristic for the DFLP is indeed capable in performing better than all other comparing evolutionary and heuristic algorithms with respect to solution quality for the test

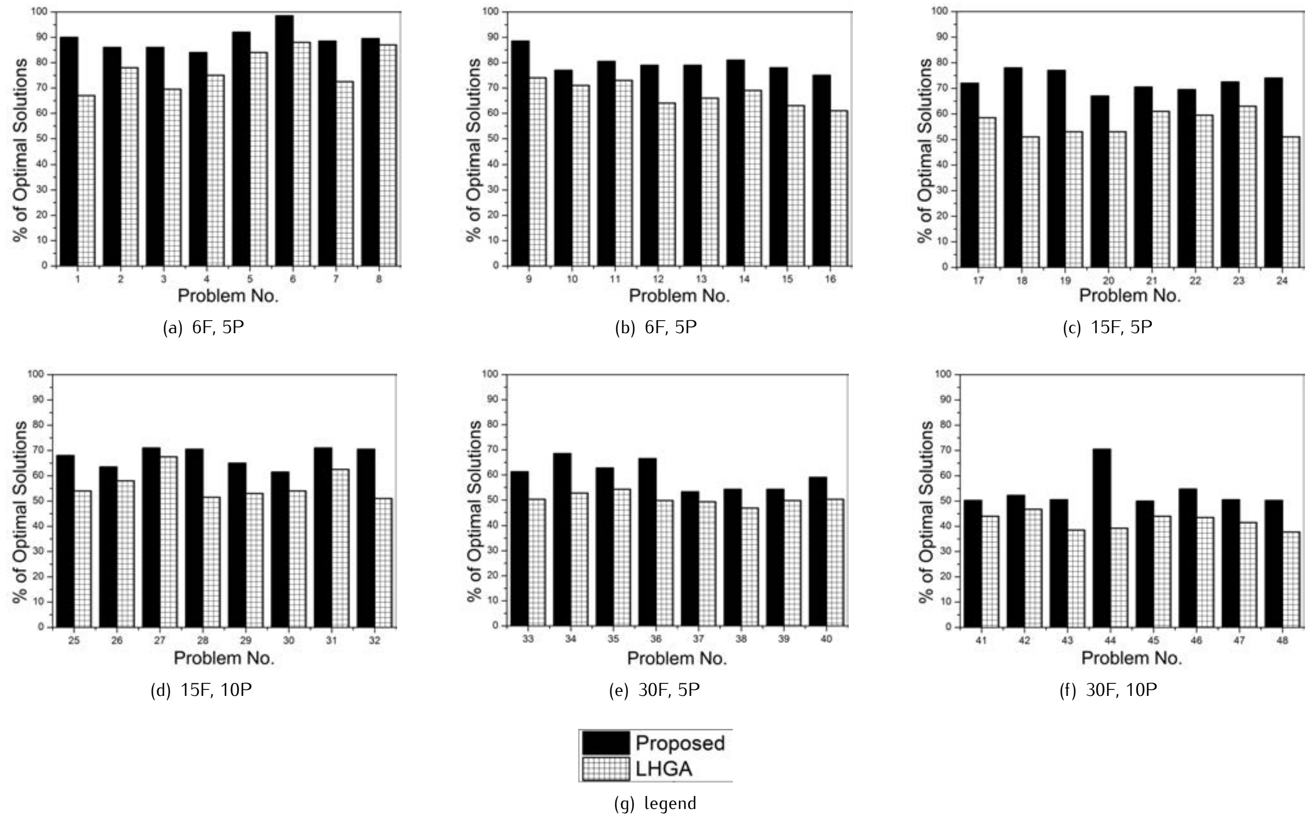


Figure 5. Percentages of optimal solution after 50% generations.

problems. It also justifies the application of the modified heuristic.

Today industrial robots present a mature technology. The huge deployment of robots and their maintenance also incur cost. Therefore, the companies aim to maximize their profit by increasing the performance of the robots and to minimize the energy used by the robots for getting better and long service from the robots. Succinctly, adapting robots to DFLPs, optimal layout for DFLPs can help the robots completing the material transporting sequence using a minimum amount of energy. It is apparent from the experimental results that the proposed approach certainly increases the effectiveness of robots by solving the DFLP, and thus saves the energy used by the robots for carrying the material among the facilities. At the same time, transporting materials among the facilities of an optimal layout for a DFLP by robots will increase the throughput of the manufacturing plant. And, higher rates of throughput mean higher profits. Consequently, it helps increasing productivity and profits for manufacturing system.

Since the algorithms used in this study use different computing systems, coding techniques and compilers, it is very difficult to compare the computation time for them. For this reason, we do not compare the computational time in this study. In future, we hope to make this comparison. Besides, there are no published benchmark data sets to compare the effects of robot as a material handling device in the DFLP. Therefore, we are unable to test this consequence. Also, in future, we would like to create some benchmark data sets or collect real-world industrial data for this comparison.

8. Conclusions

In this paper, we present a modified backward pass heuristic based genetic algorithm for solving the dynamic facility layout problem (DFLP) which can be thought of a generalized form of a framework for adapting robots to DFLPs. These days, there is an increasing trend towards implementation of industrial robot as a material handling device in manufacturing plants. Solving the DFLP and finding the optimal layout for a manufacturing system that uses robots as material handling device is an efficient and cost-effective way to improve the productivity in a plant. The experimental results strongly support the competitiveness of our proposed approach for solving the DFLP. In particular, the proposed approach has proved to be computationally efficient when dealing with large DFLPs. Furthermore, industrial robots ensure increased productivity in time critical situations when the layouts are optimal. As a consequence, the energy consumption of employed robots is minimized in material transportation between facilities, which will lead to overall cost reduction and profit maximization for manufacturing industries.

Acknowledgements

The authors would like to thank Professor J. Balakrishnan and Professor C. H. Cheng for providing their data sets for the experiment.

References

[1] S. S. Heragu, Facilities design, BWS, Boston, 1997

[2] D. Li, P. L. P. Rau, Y. Li, A cross-cultural study: Effect of robot appearance and Task, INT J SOC ROBOT, 2, (2010), 175–186

[3] K. Holck, C. Munoz, Robots in Manufacturing, Available from: http://mfg.eng.rpi.edu/gmp/SProjectsPwrpt/s11/ GMP Final Robotics Presentation.pdf

[4] H. Zhang, Factory automation with industrial robots, ARB Robotics R&D Center, 2011

[5] A. Tompkins, Facilities planning, 3rd ed., John Wiley & Sons, New York, 2003

[6] A. R. McKendall Jr, J. Shang, Hybrid ant systems for the dynamic facility layout problem, COMPUT OPER RES, 33, (2006), 790–803

[7] K. S. N. Ripon, K. Glette, M. Hovin, J. Torresen, Dynamic facility layout problem with hybrid genetic algorithm, In M. Oussalah, R. Mitchell, N. H. Siddique (Eds.), Proceedings of The IEEE 9th International Conference on Cybernetic Intelligent Systems (CIS), (2010, Reading, UK), IEEE Systems, Man, & Cybernetics Society, 2010, 33–38

[8] M. A. El-Baz, A genetic algorithm for facility layout problems of different manufacturing environments, COMPUT IND ENG, 47, (2004), 233–246

[9] Z. Jian, L. A. Ping, Genetic algorithm for robot workcell layout problem, (2009, Xiamen, China), Proceedings of The WRI World Congress on Software Engineering (WCSE 2009), 2009, 460–464

[10] K. Mitsuhashi, K. Yamato, Layout planning and simulation for application of robots, ADV ROBOTICS, 2, 1, (1987), 87–98

[11] V. Nata, A. Tubaileh, The machine layout problem in robot cells, INT J PROD RES, 36, 5, (1998), 1273–1292

[12] A. Tubaileh, I. Hammad, L. A. Kafafi, Robot cell planning, ENG TECH, 26, (2007), World Academy of Science, 250–253, 2007

[13] A. Drira, H. Pierreval, S. H. Gabouj, Facility layout problems: A survey, ANNU REV CONTROL, 31, 2, (2007), 255–267

[14] I. Castillo, T. Sim, A spring-embedding approach for the facility layout problem, J OPER RES SOC, 55, 1, (2004), 73–81

[15] J. Balakrishnan, C. H. Cheng, Dynamic layout algorithms: A state-of-the-art survey, OMEGA–INT J MGMT SCI, 26, 4, (1998), 507–521

[16] J. H. Holland, Adaptation in natural and artificial systems, The University of Michigan Press, Ann Arbor, MI, 1975

[17] M. H. Hu, M. -J. Wang, Using genetic algorithms on facilities layout problems, INT J ADV MANUF TECH, 23, 3–4, (2004), 301–310

[18] J. Balakrishnan, C. H. Cheng, D. G. Conway, An improved pair-wise exchange heuristic for the dynamic plant layout problem, INT J PROD RES, 38, 13, (2000), 3067–3077

[19] K. S. Tang, R. J. Yin, S. Kwong, K. T. Ng, K. F. Man, A theoretical development and analysis of jumping gene genetic algorithm, IEEE T IND INFORM, 7, 3, (2011), 408–418

[20] W. S. Tang, S. Kwong, K. F. Man, A jumping genes paradigm: Theory, verification and applications, IEEE CIRC SYST MANAG M, 8, 4, (2008), 18–36

[21] T. L. Urban, A heuristic for the dynamic facility layout problem, IIE TRANS, 25, 4, (1993), 57–63

[22] M. J. Rosenblatt, The dynamics of plant layout, MANAGE SCI, 32, 1, (1986), 76–86

[23] J. Balakrishnan, C. H. Cheng, The dynamic plant layout problem: Incorporating rolling horizons and forecast uncertainty, OMEGA–INT J MGMT SCI, 37, 1, (2009), 165–177

[24] S. P. Singh, R. R. K. Sharma, A review of different approaches to the facility layout problems, INT J ADV MANUF TECH, 30, 5–6, (2006), 425–433

[25] J. Balakrishnan, R. F. Jacobs, M. A. Venkataramanan, Solutions for the constrained dynamic facility layout problem, EUR J OPER RES, 57, 2, (1992), 280–286

[26] D. G. Conway, M. A. Venkataramanan, Genetic search and the dynamic facility layout problem, COMPUT OPER RES, 21, 8, (1994), 955–960

[27] J. Balakrishnan, C. H. Cheng, Genetic search and the dynamic layout problem, COMPUT OPER RES, 27, 6, (2000), 587–593

[28] J. Balakrishnan, C. H. Cheng, D. G. Conway, C. M. Lau, A hybrid genetic algorithm for the dynamic plant layout problem, INT J PROD ECON, 86, (2003), 107–120

[29] S. P. Singh, R. R. K. Sharma, Genetic algorithm based heuristics for the dynamic facility layout problem, EUR J MGMT, 8, 1, (2008), 128–134

[30] B. K. Kaku, J. B. Mazzola, A tabu search heuristic for the dynamic plant layout problem, INFORMS J COMPUT, 9, 4, (1997), 374–384

[31] J. B. Erel, J. Ghosh, J. T. Simon, New heuristic for the dynamic layout problem, J OPER RES SOC, 54, (2003), 1275–1282

[32] A. Baykasoglu, N. N. Z. Gindy, A simulated annealing algorithm for the dynamic layout problem, COMPUT OPER RES, 28, (2001), 1403–1426

[33] A. R. McKendall Jr, J. Shang, S. Kuppusamy S, Simulated annealing heuristics for the dynamic facility layout problem, COMPUT OPER RES, 33, 8, (2006), 2431–44

[34] H. Rezazadeh, M. Ghazanfari, M. S. Mehrabad, An extended discrete particle swarm optimization algorithm for the dynamic facility layout problem, J ZHEJIANG UNIV-SC A, 10, 4, (2009), 520–529

[35] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA–II, IEEE T EVOLUT COMPUT, 6, 2, (2002), 182–197

[36] K. F. Man, T. M. Chan, K. S. Tang, S. Kwong, Jumping-genes in evolutionary computing, Proceedings of The 30th Annual Conference of the IEEE Industrial Electronics Society, (2004, Pusan, Korea), IEEE Industrial Electronics Society, 2004, 1268–1272

[37] K. S. N. Ripon , K. Glette, M. Hovin, J. Torresen, Multi-objective evolutionary approach for solving facility layout problem using local search, In Proceedings of the 2010 ACM Symposium on Applied Computing, (2010, Sierre, Switzerland), ACM, New York, NY, USA, 2010, 1155–1156

[38] S. Y. Zheng, S. H. Yeung, W. S. Chan, K. F . Man, K. S. Tang, Design of broadband hybrid coupler with tight coupling using jumping gene evolutionary algorithm, IEEE T IND ELECTRON, 56, 8, (2009), 2987–2991.

[39] K. S. N. Ripon, Hybrid evolutionary approach for multi-objective job-shop scheduling problem, MALAYAS J COMPUT SCI, 20, 2, (2007), 183–198

[40] T. M. Chan, K. F. Man, K. S. Tang, S. Kwong, A jumping-genes paradigm for optimizing factory WLAN network, IEEE T IND INFORM, 3, 1, (2007), 33–43

[41] K. S. N. Ripon, S. Kwong, K. F. Man, A real coding jumping gene genetic algorithm (RJGGA) for multiobjective optimization, INFORM SCIENCES, 177, 2, (2007), 632–654

[42] A. Kusiak, S. Heragu, The facility layout problem, EUR J OPER RES, 29, (1987), 229-251

[43] L. Wang, S. Keshavarzmanesh, H. -Y. Feng, A hybrid approach for dynamic assembly shop floor layout, Proceedings of The 6th Annual IEEE Conference on Automation Science and Engineering,

(2010, Toronto, Ontario, Canada), 2010, 604–609
[44] V. Kumar, G. Bekey, Y. Zheng, Industrial, personal and service robots, In: G. Bekey (Ed.), Robotics: State Of The Art And Future Challenges (World Technology Evaluation Center, Lancaster, 2008), 89–101