

# Two-Phase Genetic Algorithm for Solving the Paired Single Row Facility Layout Problem

Hutama Parwananta, Meilinda F. N. Maghfiroh\*, Vincent F. Yu

Department of Industrial Management, National Taiwan University of Science and Technology, Taipei, Taiwan

(Received: February 1, 2013 / Revised: July 9, 2013 / Accepted: September 10, 2013)

---

## ABSTRACT

This paper proposes a two-phase genetic algorithm (GA) to solve the problem of obtaining an optimum configuration of a paired single row assembly line. We pair two single-row assembly lines due to the shared usage of several workstations, thus obtaining an optimum configuration by considering the material flow of the two rows simultaneously. The problem deals with assigning workstations to a sequence and selecting the best arrangement by looking at the length and width for each workstation. This can be considered as an enhancement of the single row facility layout problem (SRFLP), or the so-called paired SRFLP (PSRFLP). The objective of this PSRFLP is to find an optimal configuration that seeks to minimize the distance traveled by the material handler and even the use of the material handler itself if this is possible. Real-world applications of such a problem can be found for apparel, shoe, and other manual assembly lines. This research produces the schematic representation solution using the heuristic approach. The crossover and mutation will be utilized using the schematic representation solution to obtain the neighborhood solutions. The first phase of the GA result is recorded to get the best pair. Based on these best matched pairs, the second-phase GA can commence.

Keywords: Layout Problem, Paired Assembly Line, Material Handler, Genetic Algorithm

\* Corresponding Author, E-mail: [meilinda.maghfiroh@gmail.com](mailto:meilinda.maghfiroh@gmail.com)

---

## 1. INTRODUCTION

In this paper we address the problem of obtaining an optimum configuration of a paired single row assembly line. Since the two single rows are paired due to the shared usage of several workstations, the optimum configuration can only be obtained if we consider the material flow of the two rows simultaneously. This problem is a form of the facility layout problem (FLP), which already has a huge body of research as can be seen in one of the most referred review papers on FLB by Heragu and Kusiak (1987) and the newer version by Drira *et al.* (2007). More specifically, this problem deals with assigning workstations to a sequence and selecting the best arrangement that gives the minimum total material transferring distance of the products. Thus, a single row facility layout approach is considered to be the most compatible.

The problem is unique due to several factors: the unequal width of workstations, the sequential assignment, and the material transfer method. The material transfer is conducted manually by the worker assigned to the workstations and the material handler. The material handler is only used when the material transfer task cannot be done by the worker assigned to the workstations due to the arrangement that makes the origin and destination workstations not adjacent. Since using a material handler is an addition to the worker count, the optimal configuration seeks to minimize the distance traveled by the material handler and even the use of the material handler itself if this is possible. Real-world applications of such a problem can be found on apparel, shoe, and other manual assembly lines.

The rest of the paper is organized as follows. Section 2 presents a review of past research studies on this prob-

lem to give insight into the current stage of knowledge and how this research will contribute to it. Section 3 clearly defines the problem along with graphical illustrations and mathematical notations. Section 4 offers the proposed genetic algorithm (GA), including the proposed chromosome as its decision variable. Section 5 discusses a computational experiment on a case study. Finally, Section 6 concludes the paper and points out future research directions.

## 2. PREVIOUS STUDIES

The FLP concerns the physical placement of interacting facilities on a site. The FLP has an important effect on the efficiency and the profitability of manufacturing systems from the standpoint of cost and time (Ariafar and Ismail, 2009). Some modifications of FLP real-world applications have resulted in some studies looking to solve it, using the exact method, heuristic, and meta-heuristic approaches. Methods proposed for its exact solution include branch and bound, dynamic programming (Picard and Queyranne, 1981) and mixed integer linear programming. Heragu and Kusiak (1991) solved a single row facility layout problem (SRFLP) with a linear mixed integer formulation using a penalty technique. The branch and bound method is proposed to solve a two-dimensional facility layout, considering production capacity, multiple types of machines, processing route parts, and dimensions of the machines (Solimanpur and Jafari, 2008). Due to difficulties in it, some research studies have used meta-heuristic methods for solving the FLP such as Tabu search (Samarghandi and Eshghi, 2010), simulated annealing (Mir and Imam, 2001; Tam, 1992a), GA (Al-Hakim, 2000; Datta *et al.*, 2011; Tam, 1992b), particle swarm optimization (Samarghandi *et al.*, 2010), and ant colony (Hani *et al.*, 2007).

In real-world conditions, manufacturing practices normally have some constraints and or require particular layout configurations, such as single row, multiple rows, semicircular, or loop structures (Dimopoulos and Zalzal, 2000). Most research studies conducted either do not consider the dimensions of the machine/workstation or assume it to be equal (Braglia, 1996). There is also the case where the layout problem needs to consider the assembly line based on result of line balancing. The high complexity of the FLP makes the FLP problem be classified as an NP-hard problem. Thus, one may consider the meta-heuristic approach to solve it. The GA is proven to be able to solve difficult optimization problems such as SRFLP (Braglia, 1996; Datta *et al.*, 2011; Kochhar and Heragu, 1999; Sardzadeh, 2012). Our research produces the initial solution using the heuristic approach, while applying the GA procedure to improve the solution. Since this problem addresses a paired single row, the solution space becomes larger. We, therefore propose two-phase GA. In the first

phase, an initial population is randomly generated, while in the second phase the initial solution is based on the best matched pair solutions.

## 3. PROBLEM DEFINITION

The problem addressed in this paper is how to determine the optimum workstation configuration on a paired single line layout. We use the term “paired”, because the line is actually a combination of two single lines that produce a different kind of product—for illustrative purposes named products A and B—but are joined due to their shared use (working on products A and B) of several workstations. This method of sharing workstations is used to maximize the utilization of machines and workers, thus saving on a company’s production cost, and also due to the same starting point and end point of the two lines.

The optimal configuration is sought by placing  $n$  workstations and 1 aisle to a  $2(n+1)$  possible sequential layout space. If we consider the unfilled space as a workstation, then the problem equals a permutation problem with  $2(n+1)$  objects. This problem is very hard to solve with an exact optimization method—for example, if we consider only 3 workstations, the solution space is already  $8! = 40,320$  solutions. When the line consists of 27 workstations, such as in one of the problems observed, the solution space becomes  $27! = 1.09 \times 10^{28}$  solutions. Hence, this problem can be considered an NP-hard problem, because the solution space increases exponentially with the number of workstations considered.

To simplify the explanation we use the following notations throughout the paper.

$k$	: workstation ID,
$K$	: total number of workstations,
$za$	: workstation position, $za$ is 1 for upper line and 0 for bottom line,
$zb$	: workstation position (sequence on the specific line),
$W_k$	: width of workstation $k$ ,
$XB_k$	: starting $x$ coordinate of workstation $k$ ,
$XE_k$	: ending $x$ coordinate of workstation $k$ ,
$H_k$	: height of workstation $k$ ,
$YB_k$	: starting $y$ coordinate of workstation $k$ ,
$YE_k$	: ending $y$ coordinate of workstation $k$ ,
$AS_{k,za,zb}$	: workstation placement on the line, $AS_{k,za,zb} = 1$ if workstation $k$ is assigned to line $za$ and in $zb$ sequence,
$Ad_{ij}$	: adjacent status. $Ad_{ij} = 1$ if workstation $i$ is adjacent to workstation $j$ and $Ad_{ij} = 0$ , otherwise,
$MTH_{ij}$	: material handler requirement status. If a material handler is needed to transfer material from workstation $i$ to workstation $j$ , then $MTH_{ij} = 1$ , else $MTH_{ij} = 0$ ,

- $f_{ij}$  : workflow between workstations. For  $i, j \in k$ ;  $f_{ij} = 1$  if there exists a requirement to transfer a work piece from workstation  $i$  to  $j$  (if  $i = j$ , then  $f_{ij} = 0$ , and there is no transfer within the workstation),
- $d_{ij}$  : distance traveled by material handler from workstation  $i$  to  $j$ ,
- $P_k$  : penalty due to material transfer from workstation  $k$  to any destination workstation.

### 3.1 Workstation Arrangement

There are several constraints that have to be met when arranging the workstations into the available spaces or sequences. To give an illustration, we use a 10-workstation problem as can be seen in Table 1. The information given therein is workstation ID, machine type (which defines the size of the workstation), the previous workstation and next workstation, and the task ID number.

All of this information is needed to assign a workstation into the line due to several applied rules.

- 1)  $z_a$  represents the line position; if the workstation is located on the upper line, then  $z_a = 1$ ;  $z_a = 2$  for the workstation located on the bottom line.
- 2)  $z_b$  represents the order of the workstation on a specific line  $z_a$ .  $z_b$  must be in a sequential order without skipping any number. For example, a workstation cannot be assigned to a position of  $z_a = 1$  and  $z_b = 3$  if there are no workstations assigned to positions  $z_a = 1$  and  $z_b = 2$ .
- 3) The workstations cannot overlap with each other in the X axis and Y axis.
- 4) The workstations and aisle are assigned in a serial manner on the upper line or bottom line in the direction of the X axis. They are set so that there will be no empty space between the lines.
- 5) For the bottom line,  $Y E_k$  is on the X axis. Thus,  $Y B_k$  will start with a negative coordinate in accordance to the workstation height ( $H_k$ ).
- 6) When the aisle is assigned to a specific sequence, it takes spaces of the upper line and bottom line. Thus, if the workstation is violated by the aisle, then it needs to be moved after the aisle.

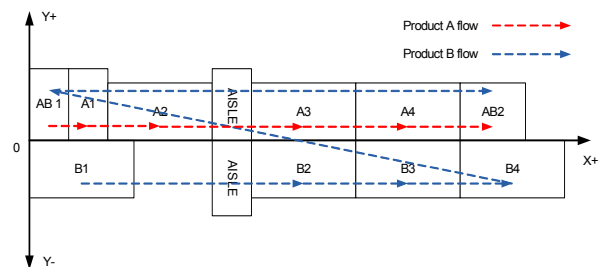
In Figure 1, we can see the assignment of workstations to the line for the illustrative problem given in Table 1. This assignment uses a basic logical assignment with product A as the main concern, without taking notice of product B's excessive material handling distance. As for the aisle placement, we use the midpoint of the upper line.

### 3.2 Material Transfer Method

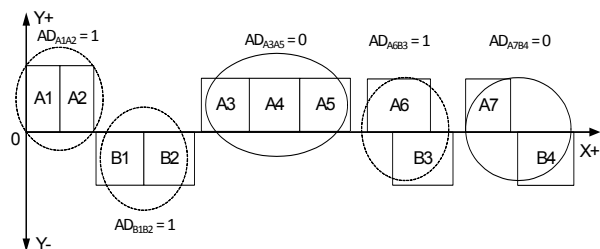
The material transfer between the workstations on

**Table 1.** Problem illustration with 10 workstations

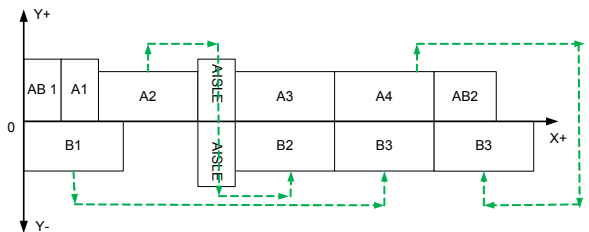
WS ID	Product	Machine type	X	Y	From	To
Start	0	0	0	0	0	B1,AB1
A1	A	I	1.5	2.5	AB1	A2,A3
A2	A	IX	3.5	2	A1	A4
A3	A	IX	3.5	2	A1	A4
A4	A	IX	3.5	2	A3	AB2
B1	B	X	3.5	2	Start	B2
B2	B	X	3.5	2	B1	B2,B3
B3	B	X	3.5	2	B2	B4
B4	B	X	3.5	2	B3	AB1
AB1	AB	IV	1.5	2.5	B4,Start	A1,AB2
AB2	AB	VI	2	2	A4,AB1	END
END	0	0	0	0	AB2	0



**Figure 1.** Workstations' assignment for the illustrative problem with 10 workstations.



**Figure 2.** Workstations' adjacent condition.



**Figure 3.** Material handler path between workstations.

the production line is done manually, and on the specific problem addressed herein there are two ways to do it. The first method is the direct transfer. The worker assigned to the origin workstation directly passes the finished material to the destination workstation after his job is finished. This method can only be done if the destination station is

adjacent to the origin station. There are two conditions that make the workstations adjacent to each other ( $Ad_{ij} = 1$ ), as can be seen in Figure 2. We assign  $i$  to represent the origin station and  $j$  as the destination station to simplify the explanation.

- 1) The origin station and destination station are next to each other on the same line, either the upper line ( $za = 1$ ) or the bottom line ( $za = 2$ ).
- 2) The origin station  $X_i$  coordinate range ( $XE_i$  to  $XB_i$ ) intersects with  $X_j$  coordinate ( $XE_j$  to  $XB_j$ ). This enables the operator at the origin workstation to transfer the material to the workstation in front of him.

The last method to transfer material between workstations uses a material handler not assigned to any workstation, or in other words, an extra worker. To transfer the material the material handler worker needs to walk from the origin workstation to the destination workstation, and if he needs to go across to the opposite side, then he must walk through the beginning of the line, the aisle, or the end of the line. We set up that the material handler will always choose the closest path. Figure 3 presents the illustration of this path. The material handler is required only when a direct transfer cannot be done.

### 3.3 Objectives' Function

We define the optimum configuration as the configuration that minimizes the material transfer penalty. The penalty only applies if the material handler commences the material transfer, and the penalties are the distances he travels from origin to destination workstation. Thus, the objective function is:

$$\min \sum_{k=1}^m P_k \quad (1)$$

The penalty only applies if there is a transfer between workstations  $i$  and  $j$  ( $f_{ij} = 1$ ) and that the transfer needs a material handler to do it ( $MTH_{ij} = 1$ ). The distance traveled by the material handler ( $d_{ij}$ ) becomes the value of the penalty, and the total value of the penalty is acquired by summing the penalties of all the transfers that occurred. This distance is determined by the sequential arrangement of the workstation, and thus the optimal arrangement is the one giving the minimum penalty.

$$P_k = \sum_{i=1}^k \sum_{j=1}^k f_{ij} MTH_{ij} d_{ij} \quad (2)$$

## 4. PROPOSED METHOD

As the FLP is an NP-hard problem, the proposed

formulation is not applicable to finding optimal solutions. Hence, the heuristic approaches are necessary to quickly obtain solutions to this problem. In this research we define two stages to solve the SRFLP: heuristic and meta-heuristic. The first stage, the heuristic method, obtains the initial solution as the basis. We then use a GA to improve the result of the objectives. Each chromosome/individual of the GA represents a permutation of the workstation. The chromosomes with the heuristic crossover and mutation operators are developed through some repetitions. The GA is a stochastic search technique that mimics the mechanisms of the Darwinian evolution based on the concept of the survival of the fittest (Goldberg, 1989).

The basic component of a GA is the solution representation, popularly known as the chromosome or individual, which represents a problem's complete solution. A GA begins with a set of random individuals, referred to as a population, which evolves over generations (iterations) by repeated applications of some genetic operators analogous to ones from natural evolution, such as selection, crossover, and mutation (Datta *et al.*, 2011). Figure 4 presents the flowchart of the proposed GA. The processes of the proposed algorithm are described in the following subsection.

### 4.1 Heuristic Method

The result from the heuristic approach becomes the basis for the performance measurement of the GA. To obtain the solution, this paper considers using a simple heuristic method. The simple heuristic chooses to minimize the flow of only one line, without considering the other. By doing so, we at least get the minimum flow of one product. The results of this heuristic method for each case study are presented later on in this paper. To give a better explanation, we present one problem with 10 workstations as in Figure 5. Figure 5 shows the flow between workstations for both products A and B. This matrix is represents elements of product A( $i, j$ ) and B( $i, j$ ) that contain the flow from workstation  $i$  to workstation  $j$ .

### 4.2 Genetic Algorithm

As one of the meta-heuristic methods that are proven for finding a good solution, the GA has been used for several kinds of problems. This paper proposes to use GA to solve the FLP in a manufacturing system. The proposed GA generates a set of assignment sequence as a solution representation

#### 4.2.1 Chromosome design

The solution represented by chromosome for a problem with N as the total number of workstations to assign can be seen on Figure 6. The chromosome contains N

workstation and 2 additional genes which are “split” and “aisle”.

The “split” gene is used to determine the cut-off point of top and bottom row, while the “aisle” gene resembles the aisle insertion point.

For an example, chromosome with a set of genes A1-A2-A3-A4-AB1-SPLIT-B1-B2-AISLE-B3 represent a solution which made up of A1, A2, A3, A4, and AB1

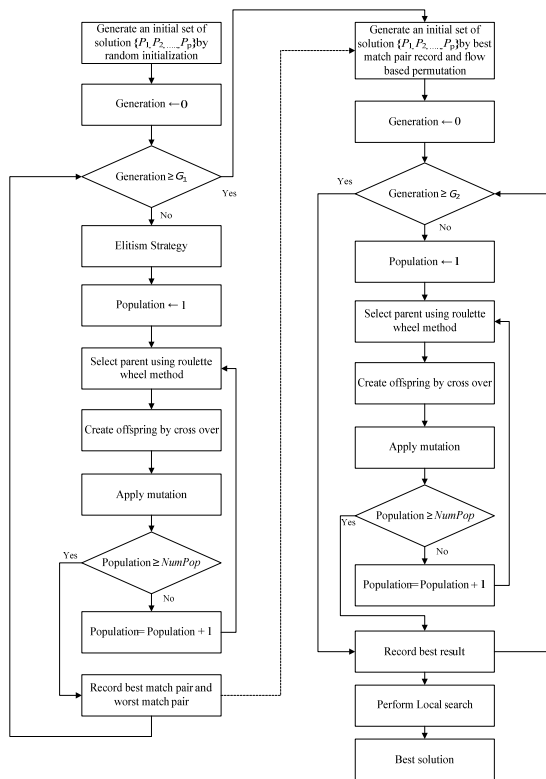


Figure 4. Flow chart of the proposed two-phase genetic algorithm.

		From to Matrix										
		A1	A2	A3	A4	B1	B2	B3	B4	AB1	AB2	END
START		0	0	0	0	1	0	0	0	1	0	0
A1		0	1	1	0	0	0	0	0	0	0	0
A2		0	0	0	1	0	0	0	0	0	0	0
A3		0	0	0	1	0	0	0	0	0	0	0
A4		0	0	0	0	0	0	0	0	0	1	0
B1		0	0	0	0	0	1	0	0	0	0	0
B2		0	0	0	0	0	0	1	0	0	0	0
B3		0	0	0	0	0	0	0	1	0	0	0
B4		0	0	0	0	0	0	0	0	1	0	0
AB1		1	0	0	0	0	0	0	0	0	1	0
AB2		0	0	0	0	0	0	0	0	0	0	1

Figure 5. From to matrix for 10 workstations.

Chromosome	n1	n2	n3	...	N	Split	Aisle
------------	----	----	----	-----	---	-------	-------

Figure 6. Solution representation.

workstations assigned sequentially on the top line and B1, B2, and B3 workstations assigned on the bottom line, and also the aisle is inserted after B2. The sequential assignment will determine the coordinates of each workstation; hence the penalty generated by material handling activity can be calculated and evaluated as chromosome’s fitness.

#### 4.2.2 Genetic algorithm procedure

The flow chart of GA procedure proposed can be seen in Figure 4. This paper proposed a two-phased GA. The first phase GA is used to record the best paired workstations in order to get a more converge random chromosome generation in the second phase. Basically the GA procedure between first phase and second phase is identical: population initialization, crossover, mutation, and reproduction. The only difference lies on the random chromosome generation constraint. Several fixed based rules are also adopted in this research for the purpose of comparison.

##### 1) Random initialization (RND)

To generate the initial population, this research uses the random base permutation method. We assign a valid random permutation to each chromosome. An  $N + 1$  gene is randomized, whereby each ID (result of permutation) represents the workstation.

##### 2) Best pair together (BPT)

We consider the pair of workstations  $i$  and  $j$  as the worst pair if the value of  $P_k$  is the highest among all pairs of workstation. In implementing the scheme, the pairs of workstation are sorted in non-increasing order of the values of their products. We then form a sub-permutation by putting together the workstations of the very first pair. For the subsequent pair  $(i, j)$ , if workstation  $i$  is already included in a sub-permutation, then put facility  $j$  on that side (left or right, and above or under) of the sub-permutation that will lead to a smaller objective value for the enhanced sub-permutation.

##### 3) Flow based permutation (FBP)

This method is based on the assumption that a good permutation will be obtained if the facilities are ordered based on the product flow. For multiple products, the permutation can be obtained from both products and compared. A permutation is then formed by putting the facilities, in order, at alternate ends of the permutation.

#### 4.2.2.1 GA-first phase

##### 1) Initialization

In the first generation of the first phase, to ensure that the solution will always be at least as good as the heuristic solution, the heuristic solution’s assignment sequence is recorded as the first chromosome. Then, random chromosomes will be generated until the population

met the parameters determined in advance.

## 2) Selection

The function of a selection operator is to form a mating, consisting of the above-average chromosomes of the population. The mating pool is used by crossover and mutation operators with the expectation of generating good offspring chromosomes. We apply the roulette wheel tournament here for this purpose. It picks up two chromosomes from the population and stores a copy of the best chromosomes (based on objective values) in the mating pool. The process is repeated until the size of the mating pool equals that of the population.

## 3) Crossover

The crossover operator is designed to generate feasible offspring chromosomes from two parent chromosomes with a predefined crossover probability (Datta *et al.*, 2011). Using a random procedure, we perform a one-point crossover. For each couple of parents with single line encoded chromosomes, a random integer is generated to choose the crossover point. The next step is to swap the range between parents 1 and 2 based on the crossover point. To make sure of the solution's feasibility, this research adopts an order crossover.

## 4) Mutation

We apply the mutation operator to create different new chromosomes and to prevent the population from stagnating in their local optimal solution with a predefined mutation probability. For this research, the mutation is done using swapping mutation. Using this random procedure, two random integers are generated as the replacing genes.

## 5) Reproduction and combination

We now combine the  $P$  chromosome of the population with the offspring chromosome created in subsection 2.2.4. The combination of the new generation is based on some rules, where 25% of the population is elite solutions, 50% are resulted from the crossover and mutation, and the last 25% are randomly generated new solutions. The elite population is to maintain the best solution recorded and it has a better solution, while randomization is performed in order to gain a different solution in the search space.

### 4.2.2.2 GA-second phase

#### 1) Best pair recording

The best pair is determined by the recorded best solution of each generation and also the two parameters best-pair together threshold (bpt-thres). The first step is to count the frequency of a pair of workstation appearing on the total generation. If result of the frequency divided by

total generation in GA first-phase is over the bpt-thres then the pair is recorded as the best pair.

## 2) Random generation

The random generation in GA-second phase is bind with the best pair result. When a new random chromosome is generated the adjacent pair within it will be matched with the best pair record. The first step is to determine the parameter of percentage of best pair match (%bpm). If the matched number divided by the total number of workstations is larger than the %bpm then the random result is accepted and inserted into the population.

### 4.2.3 Stopping criteria

We now select the maximum number of generations (G) as the stopping criteria. In this process from one generation to the next generation, we repeat the crossover and mutation until the maximum number of generations is satisfied. The stopping criterion for the first-phase GA impacts the result in the second level.

## 5. RESULTS AND DISCUSSION

### 5.1 Parameters' Setting

The effect of many different parameters on the performance characteristic in a condensed set of experiments can be examined by using the orthogonal array experimental design proposed by Taguchi. Once the parameters affecting a process that can be controlled have been determined, the levels at which these parameters should be varied must be determined. Determining what levels of a variable to test requires an in-depth understanding of the process, including the minimum, maximum, and current values of the parameter. If the difference between the minimum and maximum values of a parameter is large, then the values being tested can be further apart or more values can be tested. If the range of a parameter is small, then a smaller value can be tested or the values tested can be closer together.

Some tuning parameters used in the proposed GA are determined by using the Taguchi method. In GA, three parameters are considered: population size, crossover rate, and mutation rate. The following combinations of the parameter values were tested. Population size is 20, 40, and 60; crossover rate is 0.95, 0.90, and 0.85; and mutation rate is 0.1, 0.01, and 0.05. For each combination of tuning parameters, ten independents running are conducted for small instances with 15 work stations. The Taguchi method is conducted in Minitab 16. The result for GA's parameters is shown in Figures 7.

Based on Figure 7, the results indicate the best solution quality was obtained by setting the parameters given in Table 2.

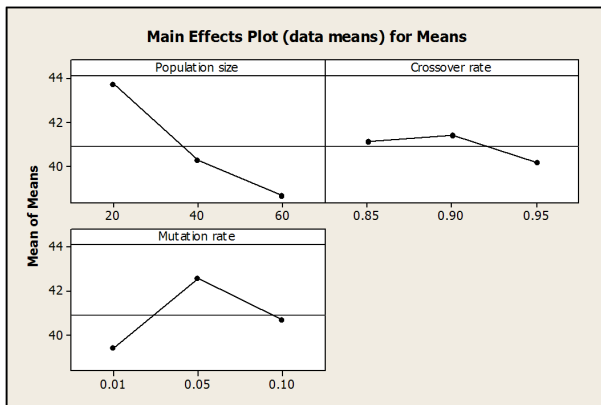


Figure 7. Main effect plot for genetic algorithm parameters using Taguchi method.

Table 2. Parameters setting for two-phase genetic algorithm

Factor	Parameter setting
Population size	60
Crossover rate	0.95
Mutation rate	0.01

Table 3. Best results obtained by two-phase genetic algorithm

Instance ID	No. of WS	Best result	Average	SD
HM_1	10	22.75	22.750	0
HM_2	15	41.25	42.600	1.244990
HM_3	20	69.25	71.500	2.883141
HM_4	25	91.75	93.000	0.935414
HM_5	27	94.00	94.625	0.883883
HM_6	30	110.50	110.875	0.530330

WS: workstation, SD: standard deviation.

Further, an extensive computational testing was performed to determine the appropriate values of others experimental parameters, e.g., threshold best pair (%bpm), generation phase-1 and generation phase-2. The results indicated that the best solution quality was obtained by setting %bpm is 80%, generation phase-1 is 10,000, and generation phase-2 is 500.

## 5.2 Computational Study

We select the maximum number of generations (G) as the stopping criteria. The proposed algorithm is coded in C# and it is executed on an Intel Corei7 2600 with 3.4 GHz processor and 4 GB RAM. This section solves for, small and medium instances using the proposed approach to evaluate the effectiveness of the proposed approach.

These instances are based on a real case problem of

a shoe manufacturing company. We handle the small instances with 10, 15, and 20 workstations, while the medium instance has 25, 27, and 30 workstations. Their optimal solutions are unknown. The results are compared with the arrangement of machines based on the heuristic approach.

### 5.2.1 Average performance by the proposed two-phase GA

We stated before that the performance of a numerical optimizer may depend upon its parameter settings. The performance of the proposed GA may vary with its parameter values, like initial solutions, population size, crossover probability, mutation probability, maximal generation for the first phase of the GA, and the threshold for the best matched pair. Therefore, in order to analyze the average performance of the GA, each of the instances is solved 20 times with different sets of such GA parameter values as mentioned in Section 5.1. In each run, the GA individuals are initialized by the RND in the first phase and BPT in the second phase. The population size is fixed at 60 in different runs. The crossover probability is set to 95% and the mutation probability is set to 1%. We set the maximal generation for the first phase at 10,000 iterations while the second phase has 500 generations. The performance of the GA is evaluated in terms of the standard deviation in objective values over 20 runs of each instance, and the average number of objective functions required in obtaining the best solution of each run. Table 3 provides the average objective values, the standard deviations, and the best result, respectively.

We observe in Table 3 that the proposed GA could obtain a good solution for each of the instances in every run (out of 20 runs). However, the standard deviation for this becomes larger with the increasing problem size. Figure 7 shows that the solution is 'near' good and the convergence point happens quite fast. As stated above, such an unavoidable variation in the performance of a meta-heuristic is quite common. Since the objective of using the meta-heuristic is to improve the line balancing result, we thus compare the result with the heuristic result that is performed using the simple heuristic approach and a single-phase GA without considering the best matched pair. The GA parameter is the same as stated in Section 5.1.

### 5.2.2 Comparison

We finally compare the obtained best solutions of Table 3 with other methods, both heuristic and meta-heuristic. We employ the heuristic method (base solution) and GA for a comparison since these two methods are the basic methods of two-phase GA. The optimum objective values obtained by heuristic for all instances are in Table 4 column 3, where the optimum objective values obtained by GA are in column 4.

**Table 4.** Comparing the solutions based on genetic algorithm (GA) and the heuristic approach

Instance ID	No. of WS	Heuristic	GA	Gap 1 (%)	Two-phase GA					
					FBP	Gap 2 (%)	BPT	Gap 3 (%)	FBP+BPT	Gap 4 (%)
HM_1	10	75.25	40.25	-46.51	28.00	-62.79	<b>22.75</b>	-69.77	<b>22.75</b>	-69.77
HM_2	15	83.00	68.50	-17.47	68.50	-17.47	57.75	-30.42	<b>41.25</b>	-50.29
HM_3	20	115.25	75.25	-34.71	75.25	-34.71	74.75	-35.14	<b>69.25</b>	-39.91
HM_4	25	218.00	135.75	-37.73	144.25	-33.83	127.25	-41.63	<b>91.75</b>	-57.91
HM_5	27	247.75	181.25	-26.84	176.50	-28.76	114.75	-53.68	<b>94.00</b>	-62.06
HM_6	30	276.50	236.00	-14.65	182.25	-34.09	149.25	-46.02	<b>110.5</b>	-60.04
Average				-29.65		-35.27		-46.11		-56.66

WS: workstation, FPB: flow based permutation, BPT: best pair together.

We tried to determine how the fixed based rule will affect to the solution by setting the different rules as the initial solution. The result for FBP, BPT, and the combination of both rules are shown in Table 4 and represented in columns 6 and 8, respectively. The result of BPT obtained by Parwananta *et al.* (2012). The proposed method, which is the combination of FBP and BPT, is presented in column 10.

In this real case problem with 6 instances, we note that the proposed GA could significantly improve the heuristic solutions (the objective values improved in the present work are shown in boldface).

Compared with the solutions obtained by the heuristic method and GA, by using the parameters in Section 5.1 the proposed two-phase GA can produce better solutions. We also see that two-phase GA outperforms the heuristic and GA procedures in terms of solution quality. The average relative percentage gap is -56.66%, obtained by dividing the difference between the two-phase GA solution and the heuristic solution by the best-known solution based on heuristic values.

Nevertheless, the meta-heuristic result outperforms the heuristic result as shown in columns 5, 7, and 9, and the BPT and FPB techniques help GA undergo diversity in the solution space. Using BPT and FBP also helps the proposed method to converge faster. In the end, the pro-

posed solution method combining both techniques can boost the performance of the proposed method.

From a computational time point of view in Table 5, the heuristic and GA seem to work better than the two-phase GA. Out of six instances, heuristic can solved the problem with less computational time. On the other hand, the larger computational effort of two-phase GA compare to heuristic and GA is due to the double iteration number of the two-phase GA. However, its computational time is reasonable for such a strategic problem that does not need to be solved every day.

## 6. CONCLUSION

In this paper we have proposed two-phase GA for solving the paired single row facility layout problem (PSRFLP), which asks for an arrangement of a number of facilities on two lines in a row with a minimum penalty caused by the material handler. The solution initialization techniques as well as the crossover and mutation operators of the proposed GA generate only valid solutions of the PSRFLP. We investigated the proposed GA on 6 instances, successfully obtaining a good solution value for each of the instances in every run (out of 20 runs) compared to the heuristic and single-phase GA results.

**Table 5.** Comparing the computational time based on GA and the heuristic approach

Instance ID	No. of WS	Running time (s)		
		Heuristic	GA	Two-phase GA
HM_1	10	1.2	13.0	46
HM_2	15	2.0	23.3	62
HM_3	20	3.2	36.1	79
HM_4	25	4.9	50.9	88
HM_5	27	5.2	56.2	153
HM_6	30	6.0	73.0	288

GA: genetic algorithm, WS: workstation.

## REFERENCES

- Al-Hakim, L. (2000), On solving facility layout problems using genetic algorithms, *International Journal of Production Research*, **38**(11), 2573-2582.
- Ariafar, S. and Ismail, N. (2009), An improved algorithm for layout design in cellular manufacturing systems, *Journal Manufacturing Systems*, **28**(4), 132-139.
- Braglia, M. (1996), Optimization of a simulated-annealing-based heuristic for single row machine layout problem by genetic algorithm, *International Transaction in Operation Research*, **1**(3), 37-49.
- Datta, D., Amaral, A. R. S., and Figueira, J. R. (2011),



- Single row facility layout problem using a permutation-based genetic algorithm, *European Journal of Operational Research*, **213**(2), 388-394.
- Dimopoulos, C. and Zalzal, A. M. S. (2000), Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons, *IEEE Transactions on Evolutionary Computation*, **4**(2), 93-113.
- Drira, A., Pierreval, H., and Hajri-Gabouj, S. (2007), Facility layout problems: a survey, *Annual Reviews in Control*, **31**(2), 255-267.
- Golberg, D. E. (1989), Genetic algorithms in search, optimization, and machine learning, *Addison-Wesley Longman Publishing Co., Inc.*
- Hani, Y., Amodeo, L., Yalaoui, F., and Chen, H. (2007), Ant colony optimization for solving an industrial layout problem, *European Journal of Operational Research*, **183**(2), 633-642.
- Heragu, S. S. and Kusiak, A. (1987), The facility layout problem, *European Journal of Operational Research*, **29**(3), 229-251.
- Heragu, S. S. and Kusiak, A. (1991), Efficient models for the facility layout problem, *European Journal of Operational Research*, **53**(1), 1-13.
- Kochhar, J. S. and Heragu, S. S. (1999), Facility layout design in a changing environment, *International Journal of Production Research*, **37**(11), 2429-2446.
- Mir, M. and Imam, M. H. (2001), A hybrid optimization approach for layout design of unequal-area facilities, *Computer & Industrial Engineering*, **39**(1-2), 49-63.
- Parwananta, H., Maghfiroh, M., and Yu, V. F. (2012), Genetic algorithm for solving paired single row facility layout problem, *Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference*, Phuket, Thailand, 151-159.
- Picard, J. C. and Queyranne, M. (1981), On the one-dimensional space allocation problem, *Operations Research*, **29**(2), 371-391.
- Samarghandi, H. and Eshghi, K. (2010), An efficient tabu algorithm for the single row facility layout problem, *European Journal of Operational Research*, **205**(1), 98-105.
- Samarghandi, H., Taabayan, P., and Jahantigh, F. F. (2010), A particle swarm optimization for the single row facility layout problem, *Computer & Industrial Engineering*, **58**(4), 529-534.
- Sadrzadeh, A. (2012), A genetic algorithm with the heuristic procedure to solve the multi-line layout problem, *Computers & Industrial Engineering*, **62**(4), 1055-1064.
- Solimanpur, M. and Jafari, A. (2008), Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm, *Computers & Industrial Engineering*, **55**(3), 606-619.
- Tam, K. Y. (1992a), A simulated annealing algorithm for allocating space to manufacturing cells, *International Journal of Production Research*, **30**(1), 63-87.
- Tam, K. Y. (1992b), Genetic algorithms, function optimization, and facility layout design, *European Journal of Operational Research*, **63**(2), 322-346.