Jurnal Teknik Industri, Vol. 23, No. 2, December 2021
ISSN 1411-2485 print / ISSN 2087-7439 online

DOI: 10.9744/jti.23.2.171-182

Solving Multi-Objective Paired Single Row Facility Layout Problem using Hybrid Variable Neighborhood Search

Meilinda Fitriani Nur Maghfiroh[1*]

Abstract: The footwear industry is distinguished by its manual assembly line and a high proportion of shared workstation configuration. This study focuses on a subset of the single row facility layout problem known as the paired single row facility layout problem. As one of type of single-row facility layout, the paired single row facility layout problem cannot be solved quickly. Further, different objectives also need to be considered in the decision-making process. Therefore, multi-objective approaches are proposed to minimize the penalty of material handler usage while maximizing the adjacency function based on each workstation's closeness rating. A Single Row Facility Layout is an NP-hard problem; this problem also belongs to the NP-hard problem class. As a result, we propose a hybrid method combining variable neighborhood search (VNS) and genetic algorithm (GA) to solve the problem of obtaining the optimal configuration of a multi-objective paired single-row assembly line. A heuristic approach was used to create the schematic representation solution. To obtain the neighborhood solutions, a hybrid VNSGA was used. The schematic representation solution employs crossover and variable neighborhood descent. Using the concept of VNS, the neighborhood was changed in each generation.

Keywords: Adjacency, material handler, multi-objectives, paired single-row layout problem, variable neighborhood search.

Introduction

Many researchers have studied the Single Row Facility Layout Problem (SRFLP) over this decade. The problem deals with configuring facilities with multiple products and minimizing the sum of the distances between all facilities [1, 2]. As it is simultaneously related to the physical placement of interacting facilities on a manufacturing line, its decision will affect the efficiency and profitability of manufacturing systems from the standpoint of cost and time [3]. FLP is an important problem in manufacturing processes, and when it is solved, the company can maximize the effectiveness of the whole operation. FLP involves arranging n unequal-area facilities of different sizes within a given total space, which can be bounded to the length or width of site area in such a way as to minimize the total material handling the cost and slack area cost [4]. Since introduced by Heragu and Kusiak [5], some modifications of FLP have been made for real case applications, as the layout problems addressed are strongly dependent on the specific features of manufacturing systems. Manufacturing practices normally have certain specific constraints: requiring certain layout configurations, such as a single row, multiple rows, semi-circular, or a loop structure [6]; various types of products and facility shapes [7]; and a sequential assignment rule. However, most research does not consider machine/ workstation dimensions or assume them equal [8] when each workstation has its specific dimensions.

The single-row layout problem occurs when facilities must be placed along a line [9,10] due to material handling devices. Many researchers have tried to solve SRFLP using different approaches, such as dynamic programming [11], mixed-integer linear programming [12], and branch and bound programming [13]. However, the high complexity of FLP has resulted in the FLP problem being classified as an NP-Hard problem. Due to its difficulties, some researchers have considered using a heuristic approach for solving FLP, such as Taboo search [14], simulated annealing [15, 16], genetic algorithm [17,18,19], particle swarm optimization [20], Lin and Kernighan heuristic [21], Hybrid EDA [1] and ant colony [22]. Several researchers have also done a study regarding multi-objective facility layout. Minimizing material handling cost, minimizing backtracking number, maximizing adjacency/ workstation closeness, maximizing distance requests, or maximizing aspect ratio requests are some of the objectives [23, 24, 25, 26, 27].

The systematic change of the neighborhood within the local search to evade local optima traps is suggested by Hansen & Mladenovic [28]. This leads to the design of a meta-heuristic approach called variable neighborhood search (VNS), further developed in various

---

[1] Faculty of Industrial Technology, Industrial Engineering Department, Universitas Islam Indonesia, Jl. Kaliurang Km 14.5, Sleman Yogyakarta 55584, Indonesia
Email: meilinda.maghfiroh@uii.ac.id

*Corresponding author

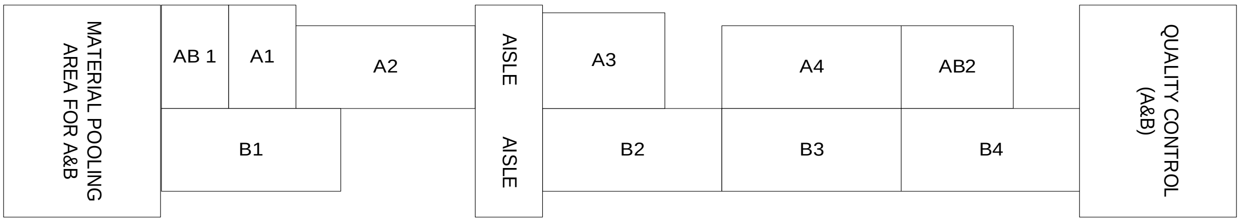

Figure 1. Example of line configuration

Table 1. Problem illustration with 10 workstations

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

extensions. VNS algorithm has been proven to solve several NP-hard and NP-complete problems such as vehicle routing [29], location routing [30], and facility layout [31]. However, this algorithm is yet to be applied for solving SRFLP. On the other hand, a genetic algorithm (GA) is proven to be able to solve difficult optimization problems such as an SRFLP [8] [19, 32, 33].

Footwear companies have become competitive players in a price-sensitive market as a rapidly growing industry. Consequently, a company's capacity to reduce production costs has become essential. One factor affecting the production cost comes from the manufacturing line configuration. Line balancing problems might be able to solve the assignment task, but the company also needs to adjust facility/workstation placement for an optimal layout. However, different from the general industry, a manufacturer with manual assembly lines such as the footwear industry must deal with a shared facility/workstation configuration. The needs of shared workstations are attributable to the execution of a similar task for producing two different products (in this problem, products A and B) with the combination of two single lines. Therefore, the optimum layout can only be obtained if the material flow of multiple rows is considered simultaneous.

The facility layout makes it possible to handle materials in an orderly, efficient manner; thus, the problem is related to arranging workstations along the material transfer path. The material transfer is conducted manually and is performed by a worker assigned to the workstation. If manual material transfer by a worker is not possible, a material handler is used. This method is usually employed if the origin and destination workstations are not adjacent. Due to its paired terminology, the problem proposed by Parwananta *et al.* [34] is called the Paired Single Row Facility Layout Problem (PSLFLP). In this model, several factors typically found in a real case situation are considered, such as unequal width and height, limited space of the assembly factory, and the sequential assignments that must be followed. However, in the previous research, the adjacency function by considering workstation relation has not been considered yet. Thus, the objectives of this research are as follows:

a) To develop and provide a complement multi-objective mathematical model of PSRFLP and solve PSRFLP problem based on a real case problem. Two objective functions, namely minimizing penalty of the material handler and maximizing adjacency function, are tackled. Small, medium, and significant size cases are considered, referring to the number of workstations present.
b) To develop hybrid Variable Neighborhood Search with a Genetic Algorithm (VNSGA) and apply it to solve SRFLP, PSRFLP, and multi-objective PSRFLP. By solving SRFLP effectively, it is hoped that an algorithm can also succeed in solving the different cases of the Facility Layout Problem.

Methods

Problem Definition

The facility layout problem in this study is focused on finding the best facility layout to minimize the total cost of transporting materials between workstations and maximizing adjacency function while at the same time satisfying the constraints of areas, aspect dimensions of the workstations, and the condition of workstation sharing.

This method of sharing a workstation maximizes the utilization of equipment and workers, thus saving production costs. It is also chosen due to the same starting and endpoints of the two lines. The problem addressed is determining the optimum workstation configuration in a two single lines layout with sharing workstation, so-called “paired-single line layout”.

Figure 1 shows the example of a 10-workstation configuration after performing line balancing. As an illustration, we will use this 10-workstation problem example from Parwananta *et al.* [35] as seen in Table 1.

The table gives information such as workstation ID, product name, machine type (which defines the task of each workstation), workstation size (x-axis shows length and y-axis corresponds to width), previous workstations (From), and next workstations (To), based on a precedence chart, and adjacency request. The WS ID column mentions the ID of each workstation. “Start” is a starting point where all materials for both products come from, while END means the endpoint (in this case, the quality control station). Each workstation performs a particular task for the selected product based on information product in column 2 and machine type in column 3. All operations are based on the precedence diagram, as shown in columns 6 and 7. The last column indicates that the particular workstations must be adjacent to the workstation before (column 6).

The optimum configuration is defined as the configuration that minimizes the penalty. The penalty applies if the material handler commences the material transfer, and the penalties are defined as the distances the material handler traveled from the original to the destined workstation. Following Parwananta *et al.* [35] with additional information, we will use notations to simplify the explanation and mathematical details throughout the paper.

k : Workstation ID
K : Total number of workstations
L : Total number of locations
α : Workstation position; $\alpha = 1$ if workstation is on the top line and $\alpha = 2$ if workstation is in the bottom line
β : The order of the workstation on a specific line α
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
mid_X : The middle point X coordinate of workstation k in a line sequence
$AS_{k,\alpha,\beta}$: Workstation placement on the line; $AS_{k,\alpha,\beta} = 1$ if workstation k is assigned to line α and in β sequence
Ad_{ij} : The adjacent status. $Ad_{ij} = 1$ if workstation i is adjacent to workstation j, $Ad_{ij} = 0$, otherwise
r_{ij} : The closeness score between workstation i to be adjacent to workstation j.
MTH_{ij} : Material handler requirement status. If a material handler is needed the transfer material from workstation i to workstation j, then $MTH_{ij} = 1$, or else $MTH_{ij} = 0$.
YTR : The maximum value of $|YB - YE|$ for the top line
YBR : The maximum value of $|YB - YE|$ for the bottom line
f_{ij} : Workflow between workstations. For $i, j \in k$; $f_{ij} = 1$ if there is 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 a workstation.
d_{ij} : Distance travelled by a material handler from workstation i to j
A_k : The adjacency value between workstation k
BT_{ij} : Backtrack status between workstations. For $ij \in k$; $BT_{ij} = 1$, if the position of workstation i to j is the opposite of the material workflow, and $BT_{ij} = 0$ if otherwise.
P_k : Penalty due to material transfer from workstation k to any destined workstation.

Workstation Arrangement

Several rules were introduced for assigning a workstation into the line. First, the arrangement of the workstations was made based on the flow of product B as the primary concern with consideration of product A's flow through the additional aisle. As for the aisle placement, we used the midpoint of the upper line (arranged workstations configuration). Finally, the constraints for the workstation arrangement are based on the assignment problem with the decision variable:

$$\sum_{l=1}^{L} x_{kl} = 1, \forall k \in K \quad (1)$$

$$\sum_{k=1}^{K} x_{kl} = 1, \forall l \in L \quad (2)$$

where, $X_{kl} = 1$ if workstation k is assigned to location l, or 0 otherwise.

Constraint (1) ensure that each workstation is assigned exactly once, while constraint (2) ensure that each location is allocated with 1 workstation.

Material Transfer Method

The material transfer between the workstations is done manually with two ways based on specific problem addressed.

Direct Transfer

Direct transfer method will be conducted if the destination workstation is adjacent to the origin workstation. In this case, the worker from the origin workstation transfers the finished material directly to the destined workstation. There are two conditions which will position the workstations adjacent to each other ($Ad_{ij} = 1$), as displayed in Figure 2.

In the case where the original workstation and destined workstation are next to each other on the same line, either in upper line ($\alpha = 1$) or bottom line ($\alpha = 2$), the adjacent status can be illustrated in Equation (3).

$$Ad_{ij} = 1\ if \begin{cases} AS_{i,\alpha,\beta}=1\ \&\ AS_{j,\alpha,\beta+1}=1 (j \text{ on the right of } i) \\ AS_{i,\alpha,\beta}=1\ \&\ AS_{j,\alpha,\beta-1}=1 (j \text{ on the right of } i) \end{cases}$$
$$else\ Ad_{ij} = 0 \quad (3)$$

Equation (4) will allow the operator of the original workstation to transfer the material in front of him if it is adjacent in a different line. If the original station $X_i$coordinate range (XE_i to XB_i) intersects with X_j coordinate (XE_j to XB_j) then the adjacent status will be 1, otherwise, the value will be 0.

$$Ad_{ij} = 1\ if (XB_i\ to\ XE_i) \cap (XB_j\ to\ XE_j) \neq 0;\ else\ Ad_{ij} = 0 \quad (4)$$

Material Handler

Transfer material between workstations can also be performed using an extra worker. The material handler will always choose the closest path; chosen from several options; from the original workstation to the destined workstation. The material handler is required when the direct transfer is not possible.

Objectives Function

Three different aspects of the facility layout are considered, material handling cost, back tracking condition and adjacency request. Three objectives are therefore considered:

Penalty due to material handler

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

$$Min\ \{m_1\} = Min\ P_k = \sum_{i=1}^{k} \sum_{j=1}^{k} f_{ij} MTH_{ij} d_{ij} \quad \forall\ i, \forall j \in k \quad (5)$$

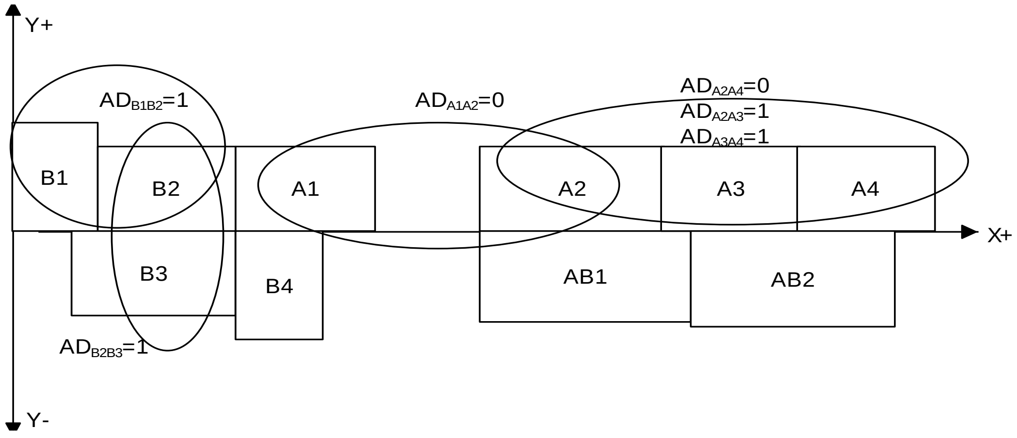

Figure 2. Workstations' adjacent condition illustration

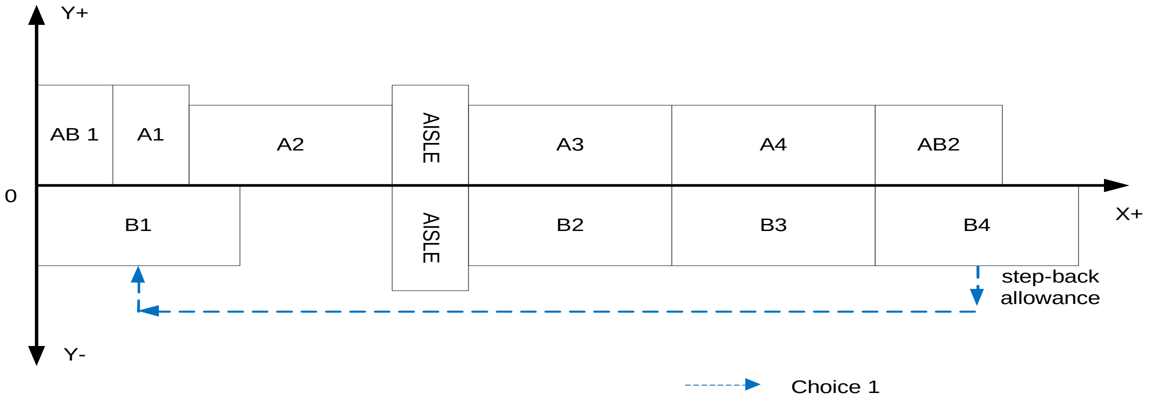

Figure 3. Material transfer within the same row

Several factors are to be considered for the calculation of the penalty; they are penalty calculation based on a two-dimensional area (X and Y); additional aisle to reduce handling; and several choices on how to move the material. The penalty calculation must be differrentiated between transferring within the same row and across two rows. If the transfer is within a row, i.e. $\alpha_{origin} = \alpha_{destination}$, then the penalty can be calculated as the distance between the centers of the workstations $(midX_k)$added by one meter as the step away tolerance, shown in Figure 3.

The center point of workstation k is calculated from its starting point on the line added by half of its width (W_k). The calculation for this penalty cost can be described in Equations (6) and (7) below:

$$midX_i = \left(XB_i + \frac{XE_i - XB_i}{2}\right) \tag{6}$$

$$d_{ij} = \left|\{mid_i - mid_j\}\right| + 1 \tag{7}$$

If the two workstations are in a different row or $\alpha_{origin} \neq \alpha_{destination}$then several steps are necessary to perform to calculate the penalty.

(a) To go from top-row to bottom-row and vice versa, there are three passable Y paths: before the first workstation, through the aisle, and after the latest workstation. Since the first is always static at (0-0.5), we need only to determine the X coordinate of the aisle and the end of the line +0.5 allowances. For ease of explanation, we name the path before workstations as path S and the path after workstations as path E.

(b) Determine the maximum value of $|YB - YE|$ for the top row and bottom row, denoted respectively as YTR and YBR.

The distance travelled which is equal to penalty thus can be calculated as follows

$$d_{ij} = Y_{TR} + Y_{BR} + min\left(midX_i \rightarrow Ypath \rightarrow midX_j\right) \\ +allowance \tag{8}$$

$$d_{ij} = Y_{TR} + Y_{BR} + \\ \min\begin{pmatrix} \left|(midX_i - X_{pathS}) + (midX_j - X_{pathS})\right|, \\ \left|(midX_i - X_{aisle}) + (midX_j - X_{aisle})\right|, \\ \left|(midX_i - X_{pathE}) + (midX_j - X_{pathE})\right| \end{pmatrix} \tag{9}$$

Adjacency Function

The second objective considering the request from the operator due to continues process during the assembly process. The adjacency function is calculated by the adjacency status and closeness score based on its closeness rating for workstation relation to another workstation.

$$Min\,\{m_2\} = \text{Min}\ \frac{1}{A_k} = \frac{1}{\sum_{i=1}^{k}\sum_{j=1}^{k} Ad_{ij}r_{ij}} \quad \forall i, \forall j \in k \tag{10}$$

According to Muther [36], closeness ratings indicate the desired "closeness" for two departments/ workstations to be next to each other. The closeness rating is being used during the development of activity relationship chart to evaluate the qualitative aspect of the facility layout. This study adopted the concept, and each rating is converted into numerical values. According to Chen [37], the value assigned to each rating is subjective and does not have particular rule, depend on the level of importance of the closeness for each department. However, as some ratings are denoted different needs, such as A (absolutely need to be close to each other) or X (cannot be close to each other), many studies used a high deviation of number. In this study, however, followed the study conducted by Chen [37] and assigned the closeness score in such ways: 6 to A, 5 to E, 4 to I, 3 to O, 2 to U and -1 to X.

Proposed Algorithm

Following the concept of VNS, this proposed algorithm used the neighborhood changes to escape from the local optimal. On the other hand, GA, which is already proven to solve FLP, was employed to generate several solutions, and then the best match pair was recorded. The main effect of hybridizing VNS with GA is improving the convergence speed to the local optima with a high diversity of solutions. Some researchers have already used the hybridization of these two methods by adding VNS as the local search [38, 39, 40]. Within a hybrid of GA and local search, VNS/VND was added to GAs and applied to every child/chromosome before it was inserted into the population.

This study developed a new approach hybridizing the VNS concept and GA to exploit both the global searchability and the local search ability for solving SRFLP and PSRFLP. The proposed VNSGA generated a set of permutations as the representation solution, where the individual was a result permutation of the workstation to be arranged. First, each chromosome of GA represented a permutation of the workstation, both for a single and shared workstation. Next, the chromosomes with the crossover operators were developed through several repetitions. VNS then started with evolving chromosomes and a set of neighborhoods $\mathrm{Nneighbor}$, $\mathrm{k} = 1, 2, \ldots, \mathrm{Nmax}$.

A solution for each chromosome was computed at each iteration with respect to the nth neighborhood, $\mathrm{Nn(sol)}$. If a new solution was found better than the current solution, then the solution was updated, and the process continued by putting a new solution into the next generation; otherwise, the same steps were repeated with the next neighborhood, $\mathrm{Nn} + 1$ until $\mathrm{n} = \mathrm{Nmax}$. Each chromosome memorized the neighborhood structure (*n*) and was used for the next generation. The difference between the basic VNS here was the neighborhood changes system. The following subsection describes the processes of the proposed algorithm.

Solution Scheme

The solution of this problem was a set of permutations of strings containing total workstations+2. For the solution scheme, the additional 2 bits was employed to represent the aisle and the split. The aisle represented the location of the aisle, while the split functioned as a splitting procedure of the top and bottom rows. Figure 4 shows the solution scheme for a workstation.

Chromosome	A1	A2	A3	A4	B1	B2	B3	B4	AB1	AB2	.	.	.	.	.	.	.	AISLE

Figure 4. Solution representation

Illustration of Solution Representation

For example, we used 10 workstations and one aisle based on the data in Table 1. Each chromosome representing the solution was split into two rows: the top ($\alpha = 1$) and bottom row ($\alpha = 2$). This splitting method used a randomized method, meaning the split can occur at any point of the solution scheme and was performed only to calculate the objective value. The solution scheme was kept in its original form when the neighborhood operators, such as crossover and shaking, were performed.

Initial Population

Once the type of individual representation was chosen, the next step was to initialize an individual. As GA worked with population, a proportion of the population was a result of heuristic approach. The approach employed product's assignment sequence, to ensure the solution can be satisfied at least one type of product. For fulfilling the population, random chromosomes were then generated for obtaining a diversified population.

Selection

The roulette wheel tournament method took two chromosomes from the population and stored a copy of the best chromosomes in the mating pool until the number of populations was filled. The mating pool would have a higher average fitness than the average population fitness.

Crossover

Two random integer numbers were generated to choose a two-point crossover for each chromosome defined as the parent. Then based on the crossover point, we swapped the range between Parent 1 and 2. To ensure the feasibility of the solution, this research adopted order crossover, meaning that a portion of one parent was mapped to a portion of another parent. From the replaced portion, onward, the rest was filled up by the remaining genes, omitting the already-present genes. The order was then preserved by the sub-sequence.

Shaking

The concept of VNS was applied after the offspring was generated. The neighborhood operator was implemented to create different new chromosomes and to prevent the population from stagnating in their local optimal solution with a predefined neighborhood structure. The proposed algorithm incorporated three neighborhood structures to explore different possibilities of workstation position. The neighbourhoods used for the algorithm is presented in Table 2.

Table 2. Neighborhoods of VNSGA

	n	Neighborhood
N_1	1-2	Random insertion
N_2	3	2-opt
N_3	4-5	Random swap

The neighborhood N_1 corresponds to the respective insertion moves. The neighborhood operator N_1 consists of removing a workstation from position *i* and inserting it after position j. The classical 2-opt operator was used for changing an edge and inserting it in another position. Using this method, it was possible to maintain the best pair solution during the shaking. Neighborhood N_3 consists of swapping the positions of two workstations in the same line or in different lines.

Record the Best Match Pair and Local Search

The algorithm was performed to obtain the best matching pair between workstations. It was expected that after multiple iterations, all the pairings bringing good performance were retained and recorded. Next, the local search phase was performed based on the best pairing result. This method bound the solution space and stabilized the algorithm. VND algorithm then applied in the pool of best pairing result to explore another possibility of the best solution.

Multi-Objective VNSGA

A modification of the proposed method is mainly executed during selection procedure and performance evaluation. Based on decision maker opinion, this study put the same weight for each objective function. We adopted NSGA II and NSGA III in the objective calculation to tackle the multi-objectives problem in this study. The NSGA-II is multi-objective GA which was first proposed by Deb *et al.* [41] to search for Pareto-optimal solutions. The process of selection is modified in which that all non-dominated solutions is shorted, and the crowding distance is calculated. For the next generation, the best solutions used is the result of crowding distance selection, as explained in Figure 5.

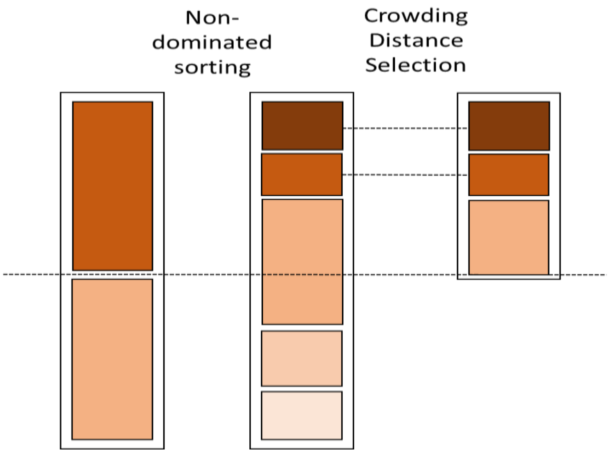

Figure 5. Modification on selection process in multi-objectives VNSGA

The multi-objectives problem can be described as

$$\min m_i(x)\,, i = 1{,}2, x \in I \tag{11}$$

with I as the feasible solution space

The non-dominated solutions are rectified after the objective's calculation process. Individual solutions are ranked in ascending order based on the values of their objective functions. The lower the rank, the greater the chromosome non-domination (individuals). Individuals with rank 1 are thus generally non-dominated. The number of non-dominated individuals generally increases with the number of generations as the search space expands and the chance of encountering more potential individuals expands. After that, each individual is assigned a crowding distance, with procedure as follow:

a. Let N the number of individuals in the ith front f_i
b. For each individual in front f_i, initialize distance to be zero for all individuals in $f_i(d(0))$.
c. For each objective function m in M, sort the individuals in f_i based on m
d. Assign finite distance to each individual in n as a boundary value of distance
e. For each individual x_i in n, calculate crowding distance by:

$$cd_i^k = \begin{cases} \frac{f_{i+1}^k - f_{i-1}^k}{f_{max}^k - f_{min}^k}, if\ index(x_i^k) \in [2, n-1] \\ \infty \qquad\qquad , if\ otherwise \end{cases} \tag{12}$$

$$cd_i \ = \textstyle\sum_{k=1}^{m} cd_i^k \tag{13}$$

where cd_i^k is the crowding distance on the *k*th objective function of x_i, m is the number of objective functions, index (x_i^k) is the sort index of x_i on the *k*th objective function, f_{i+1}^k is the value of objective function corresponding to the last solution on the axis of the *k*th objective function, andf_{max}^k and f_{min}^k represent the maximum and minimum values of the *k*th objective function, respectively.

Results and Discussion

This section describes the computational tests used to assess the performance of the proposed VNSGA algorithm for solving SRFLP and PSRFLP. First, we modified and used the proposed algorithm to solve the single-row facility layout problem (SRFLP) for the comparison against state-of-the-art in the single-row problem, as analysed in the references above. Later, subsequent sections discuss the extended problem, PSRFLP, including its test problems, parameter selection, and computational results of the proposed VNSGA. Finally, the result will be compared with the heuristics method, GA, and Two-Phase GA proposed by Parwananta *et al.* [35].

Table 3. Parameters' setting for VNSGA

Factor	Tuning parameter	Chosen Parameter Setting
Population Size	30, 40, 50	40
Crossover rate	0.90, 0.95, 0.99	0.95
Number of generations	2000, 5000, 10000	5000
N_{max}	3, 4, 5	5

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. Parameters used in the proposed GA are determined by using the Taguchi method. In GA, three parameters are considered: population size, crossover rate (CR), and the number of generations. While for the VNS, the maximum number of a neighborhood used in the algorithm is considered. The mutation rate was excluded due to the VNS concept. VNS act as a local search that replaces the function of the mutation ability in GA. The parameter's setting details is shown in Table 3.

Computational Study

The maximum number of generations was selected as the stopping criteria. The proposed algorithm was coded in C# and executed on Intel® Core™ i7 2600 with a 3.4 GHz processor and 4 GB RAM. This section attempted to solve small, medium, and large instances. These instances were based on actual data gained from apparel product companies. Following the previous study, we handled small instances with 10, 15, and 20 workstations, medium instances with 25, 27, and 30 workstations, and then extended it to big instances with 40, 45, 50, and 60 workstations. The optimal solutions for small and medium instances were taken from Parwananta *et al.* [35], with optimal solutions for big instances were considered unidentified. The results were then compared to the arrangement of machines based on the heuristic approach.

Average Performance by the Proposed VNSGA for solving SRFLP

The SRFLP is first solved and compared with the current result from other methodologies to evaluate the proposed method. First, we modified and used the proposed algorithm to solve the single-row facility layout problem (SRFLP) compared against state-of-the-art in the single-row problem. The modification includes solution representation, initial solution phase, and removal of additional 2 bits for represent aisle and split. Then, the objective function is modified and calculated by submission of the distances between all facility pairs multiple by traffic load for each link, and equation (1), (2), (6), and (7) was ignored.

As shown in Table 4, the proposed method can solve the entire basic SRFLP problem with 30 workstations with a decent result. From 15 instances, only one instance cannot be solved up to optimality. The records with a bold number mean the best-known solution found (Column 3). Column 5-6 shows the result of the VNSGA, for both the best result (Min) and the average (Average was calculated by dividing a total of 20 times run of the algorithm for each instance). Although the algorithm is failed to solve the instance with 30 facilities, it is still robust to solve the other problems and find the best know solution respectively.

This result is expected as the algorithm method was designed to solve the paired case in which additional bits were added to each solution. Nevertheless, from this result, we can see that the proposed method is reliable for solving basic SRFLP problems with a good result compared to the other methodologies.

Algorithm Performance for solving PSRFLP

As the algorithm can solve the basic SRFLP problem as discussed in Section 3.2.1 above, the algorithm is then employed for solving the extended problem PSRFLP. First, each instance was solved ten times with different VNSGA parameter values.

The best solutions were then compared to other methods, covering heuristic and metaheuristic ones, as shown in Table 5. The heuristic method (base solution), naive GA, and two-phase GA (TPGA) were employed to compare these two methods are the basic methods of VNSGA. Except for the heuristic method, all other algorithms are run ten times. Only the best result (minimum penalty value) is shown. The heuristic method results are displayed in column 3, while the values of the objective obtained by GA are in column 4. The two-phase GA results obtained from Parwananta, *et al.* [35] are shown in column 5, while the VNSGA results are displayed in column 6. The performance of VNSGA can be seen from the gap value shown in columns 7-9. In this real case problem with ten instances, it was observed that the proposed VNSGA could improve the solutions by producing the smaller penalty value, not only the heuristic solutions but also GA solutions with a big gap (the objective values improved in the present work are shown in bold). In bigger instances, compared to the solution performed with the heuristic method, GA, and TPGA, the proposed VNSGA produced a smaller penalty value and outperformed the heuristic and GA procedure. Out of 10 instances, the method used in

Table 4. The result obtained by VNSGA and its comparison for SRFLP

ID	Number of WS	Best Known Solution	Source	VNSGA*		
				Best Result	Average	Gap
P4	4	**638.0**	[1,2,14]	**638**	638	0.00%
LW5	5	**151.0**	[1, 14,19]	**151**	151	0.00%
N6	6	**1.99**	[1, 42]	**1.99**	1.99	0.00%
S8	8	**801**	[1,2,14]	**801**	801	0.00%
S8H	8	**2,324.5**	[1, 14, 19]	**2,324.5**	2,324.5	0.00%
S9	9	**2,469.5**	[1,2,14]	**2,469.5**	2,469.5	0.00%
S9H	9	**4,695.5**	[1,2,14]	**4,695.5**	4,695.5	0.00%
S10	10	**2,781.5**	[1, 14,19]	**2,781.5**	2,781.5	0.00%
S11	11	**6,933.5**	[1, 14,19]	**6,933.5**	6,951	0.00%
LW11	11	**6,933.5**	[1, 14,19]	**6,933.5**	6,937	0.00%
N12	12	**23.365**	[1, 14, 42]	**23.365**	24	0.00%
P15	15	**6,305**	[1,2,14]	**6,305**	6,435	0.00%
P20	20	**15,549**	[1, 14,19]	**15,549**	16,135	0.00%
P25	25	**4,618**	[1,43]	**4,618**	4,757	0.00%
P30	30	**44,965**	[1, 14,19]	46,319	46,984	3.01%
Average						0.20%

* The algorithm is run for 20 times. Best result is the minimum solution in 20 runs

Table 5. Comparison of solutions based on GA and heuristic approach

ID	Number of WS	Heuristic	GA	TPGA	VNSGA	Gap1* (%)	Gap2** (%)	Gap3*** (%)
HM_10	10	75.25	40.25	**22.75**	**22.75**	-70%	-43%	0%
HM_15	15	83.00	68.50	57.75	**25.75**	-69%	-62%	-55%
HM_20	20	115.30	75.25	95.00	**59.75**	-48%	-21%	-37%
HM_25	25	218.00	135.75	127.25	**80.25**	-63%	-41%	-37%
HM_27	27	247.80	181.25	114.75	**86.00**	-65%	-53%	-25%
HM_30	30	276.50	236.00	149.25	**93.50**	-66%	-60%	-37%
HM_40	40	798.25	324.75	243.00	**176.5**	-78%	-46%	-27%
HM_45	45	926.75	341.25	326.75	**191.75**	-79%	-44%	-41%
HM_50	50	922.50	429.00	340.75	**207.25**	-78%	-52%	-39%
HM_60	60	1341.00	736.5	620.25	**353.75**	-74%	-52%	-43%
Average						-69%	-47%	-34%

*Gap1=(VNSGA result- heuristic result)/heuristic result
**Gap2=(VNSGA result- GA result)/GA result
***Gap3=(VNSGA result- TPGA result)/ TPGA result

this study resulted in smaller penalty values for 9 cases. The average relative percentage gap is -34%, obtained by dividing the difference between the VNSGA solution and TPGA solution by the best-known solution based on TPGA. A workplace must be carefully designed for more effective and efficient workflow, while any changes often need thorough analysis and careful decision-making. It could be costly as well.

As mentioned in the previous section, the main effect of hybridizing VNS with GA is improving the convergence speed to the local optima with a high diversity of solutions. As shown in figure 2 above, VNSGA can outperform the TPGA in terms of objective function and convergence speed. During the 1000 iteration, VNSGA has reached the convergence level and optimal solution, while TP GA reaches convergence after 2000 iteration with deviation to the optimal solution. The same results were also found in the other instances, respectively.

Result of Multi-Objective PSRFLP

We run the algorithm for small instances from 10 WS to 30 WS to solve a multi-objective facility problem in a paired case. The feasible and non-dominated solutions obtained by the algorithm have given several solutions as optimal. As shown in Table 6, the algorithm can solve the small instances with 10 and 15 workstations with minimum penalty scores compared with a single objective problem. However, the algorithm failed to solve the rest data sets for minimizing penalty due to the material handler since the algorithm tried to solve two objectives simultaneously.

Further, we analyse the structure of the solutions for HM_25 obtained in Table 6. As multi-objective optimization result will be differed from single objective optimization, where each objective corresponds to different solution. To illustrate the solution, Pareto-front of non-dominated solution is presented in Figure 6.

Table 6. Result for multi-objective PSRFLP in small instances

Instances ID	Number of WS	Penalty	Closeness score
HM_10	10	22.75	16
HM_15	15	25.75	17
HM_20	20	65	22
HM_25	25	87.75	15
HM_27	27	134.75	12
HM_30	30	175.75	7

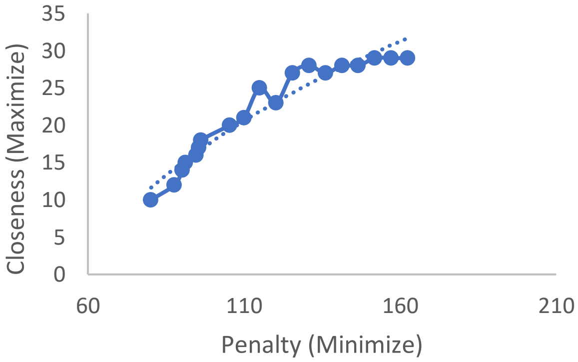

Figure 6. Pareto-front for data set HM_25

Conclusion

This paper proposes a hybrid variable neighborhood search genetic algorithm (VNSGA) for solving a particular case of SRFLP, the paired single-row facility layout problem (PSRFLP). The PSRFLP is a particular case that the facility needs to share for many products. The paired facilities require different configurations with a typical single-row facility layout. Further, we extend the problem to propose that multi-objective PSRFLP and multi-objective approaches minimize the penalty of material handler usage and maximize adjacency function based on the closeness rating of each workstation.

In conclusion, the proposed algorithm could tackle the SRFLP problem with minimum deviation. It also performed well in solving the PSRFLP with real data up to 60 workstations, with a relatively average gap of 69% compared to the heuristic method's insufficient running time. However, the algorithm still needs to be improved as it fails to solve the multi-objective problem effectively. The proposed approach can obtain the solutions certainly not optimal but can give the decision-maker a restricted number of solutions among which decision-maker can choose those considered the best.

References

1. Ou-Yang C., and Utamima, A., Hybrid Estimation of Distribution Algorithm for Solving Single Row Facility Layout Problem, *Computers & Industrial Engineering*, 66, 2013, pp. 95-103.
2. Amaral, A.R.S., A New Lower Bound for the Single Row Facility Layout Problem, *Discrete Applied Mathematics*, 157, 2009, pp. 183-190.
3. Ariafar, S., and Ismail, N., An Improved Algorithm for Layout Design in Cellular Manufacturing Systems, *Journal of Manufacturing Systems*, 28, 2009, pp. 132-139.
4. Lee, Y.H., and Lee, M.H., A Shape-based Block Layout Approach to Facility Layout Problems using Hybrid Genetic Algorithm, *Computers & Industrial Engineering*, 42, 2002, pp. 237–248.
5. Kusiak, A, and Heragu, S.S., The Facility Layout Problem, *European Journal of Operational Research*, 29, 1987, pp. 229-251.
6. Dimopoulos, C., Zalzala, A.M.S., Recent Developments in Evolutionary Computation for Manufacturing Optimization: Problems, Solutions, and Comparisons, Evolutionary Computation, *IEEE Transactions on Evolutionary Computation*, 4, 2000, pp. 93-113.
7. Drira, A., Pierreval, H., Hajri-Gabouj, S., Facility Layout Problems: A Survey, *Annual Reviews in Control*, 31, 2007, pp. 255-267.
8. Braglia, M., Optimisation of a Simulated-Annealing-based Heuristic for Single Row Machine Layout Problem by Genetic Algorithm, *International Transactions in Operational Research*, 3, 1996, pp. 37-49.
9. Djellab, H., and Gourgand, M., A New Heuristic Procedure for the Single-row Facility Layout Problem, *International Journal of Computer Integrated Manufacturing*, 14, 2001, pp. 270-280.
10. Ficko, M., Brezocnik, M., and Balic, J., Designing the Layout of Single- and Multiple-rows Flexible Manufacturing System by Genetic Algorithms, *Journal of Materials Processing Technology*, 157–158, 2004, pp. 150-158.
11. Picard, J.C., On the One Dimensional Space Allocation Problem, *Operations Research*, 29, 1981, 150-173.
12. Heragu, S.S., and Kusiak, A., Efficient Models for the Facility Layout Problem, *European Journal of Operational Research*, 53, 1991, pp. 1-13.
13. Solimanpur, M., and Jafari, A., Optimal Solution for the Two-Dimensional Facility Layout Problem Using a Branch-And-Bound Algorithm, *Computer and Industrial Engineering*, 55, 2008, pp. 606-619.
14. Samarghandi, H., and Eshghi, K., An Efficient Tabu Algorithm for the Single Row Facility Layout Problem, *European Journal of Operational Research*. 205, 2010, pp. 98-105.
15. Kar Yan T., A Simulated Annealing Algorithm for Allocating Space to Manufacturing Cells, *International Journal of Production Research*, 30, 1992, pp. 63-87.
16. Mir, M., and Imam, M.H., A Hybrid Optimization Approach for Layout Design of Unequal-Area

Facilities, *Computers & Industrial Engineering*, 39, 2001, pp. 49-63.

17. Kar Yan, T., Genetic Algorithms, Function Optimization, and Facility Layout Design, *European Journal of Operational Research*, 63, 1992, pp. 322-346.
18. Al-Hakim, L., On Solving Facility Layout Problems Using Genetic Algorithms, *International Journal of Production Research*. 38, 2000, pp. 2573-2582.
19. Datta, D., Amaral, A.R.S., and Figueira, J.R., Single Row Facility Layout Problem Using a Permutation-Based Genetic Algorithm, *European Journal of Operational Research*, 213, 2011, pp. 388-394.
20. Samarghandi, H., Taabayan, P., and Jahantigh, F.F., A Particle Swarm Optimization for the Single Row Facility Layout Problem, *Computers & Industrial Engineering*, 58, 2010, pp. 529-534.
21. Kothari, R., and Ghosh, D., Insertion Based Lin–Kernighan Heuristic for Single Row Facility Layout, *Computers & Operations Research*, 40, 2013, pp. 129-136.
22. Hani, Y., Amodeo, L., Yalaoui, F., and Chen, H., Ant Colony Optimization for Solving an Industrial Layout Problem, *European Journal of Operational Research*, 183, 2007, pp. 633-642.
23. Singh, S.P. and Singh, V. K., Three Level AHP-Based Heuristic Approach for Multi-Objective Facility Layout Problem, *International Journal of Production Research*, 49(4), 2010, pp. 1105-1125.
24. Matai, R., Singh, S.P., Mittal, M.L., Modified Simulated Annealing Based Approach for Multi Objective Facility Layout Problem, *International Journal of Production Research*, 51 (14), 2013, pp. 4273-4288.
25. Matai, R., Solving Multi Objective Facility Layout Problem by Modified Simulated Annealing, *Applied Mathematics and Computation*, 261, 2015, pp. 302-311.
26. Tayal, A., Singh, S.P., Integrated SA-DEA-TOPSIS-Based Solution Approach for Multi Objective Stochastic Dynamic Facility Layout Problem, *International Journal of Business and Systems Research*, 11(1-2), 2017, pp. 82-100.
27. Azevedo, M.M., Crispim, J.A., and Pinho de Sousa, J., A Dynamic Multi-Objective Approach for the Reconfigurable Multi-Facility Layout Problem, *Journal of Manufacturing Systems*, 42, 2017, pp. 140-152.
28. Hansen, P., and Mladenović, N., Variable Neighborhood Search for the P-Median, *Location Science*, 5, 1997, pp. 207-226.
29. Rousseau, L.M., Gendreau, M., and Pesant, G., Using Constraint-Based Operators to Solve the Vehicle Routing Problem with Time Windows, *Journal of Heuristics*, 8, 2002, pp. 43-58.
30. Maghfiroh, M.F.N, and Yu, V.F., A Variable Neighborhood Search with Path-Relinking for the Capacitated Location Routing Problem, *Journal of Industrial and Production Engineering*, 2014, pp. 1-14.
31. Ripon, K.S.N., Glete, K., Khan, K.N., Hovin, M., and Torresen, J., Adaptive Variable Neighborhood Search for Solving Multi-Objective Facility Layout Problems with Unequal Area Facilities, *Swarm and Evolutionary Computation*, 8, 2013, pp. 1-12.
32. Kochhar, J.S., and Heragu, S.S., Facility Layout Design in A Changing Environment, *International Journal of Production Research*, 37, 1999, pp. 2429-2446.
33. Sadrzadeh, A., A Genetic Algorithm with the Heuristic Procedure to Solve the Multi-Line Layout Problem, *Computers & Industrial Engineering*, 62, 2012, pp. 1055-1064.
34. Parwananta, H., Maghfiroh, M.F.N., and Yu, V.F., Genetic Algorithm for Solving Paired Single Row Facility Layout Problem, in: H.T.L. V. Kachitvichyanukul, and R. Pitakaso Eds (Ed.) *Asia Pacific Industrial Engineering & Management Systems Conference* 2012 Asian Institute of Technology, Phuket, Thailand, 2012, pp. 151-159.
35. Parwananta, H., Maghfiroh, M.F.N., and Yu, V.F., Two-Phase Genetic Algorithm for Solving The Paired Single Row Facility Layout Problem, *Journal Industrial Engineering & Management Systems*, 12, 2013, pp. 181-189.
36. Muther, R.., *Systematic Layout Planning*, 1973. Boston, CBI Publishing Company, Inc.
37. Chen, G.Y., *Multi-objective Evaluation of Dynamic Facility Layout Using Ant Colony Optimization*, The University of Texas at Arlington. ProQuest Dissertations Publishing, 2007.
38. Duda, J., A Hybrid Genetic Algorithm and Variable Neighborhood Search for Multi-Family Capacitated Lot-Sizing Problem, *Electronic Notes in Discrete Mathematics*, 58, 2017, pp. 103-110
39. Schermer, D., Moeini, M., and Wendt, O., A Hybrid VNS/Tabu Search Algorithm for Solving the Vehicle Routing Problem with Drones and En Route Operations, *Computers & Operations Research*, 109, 2019, pp. 134-158
40. Pham, D. T. and Binh Huynh, T. T., An Effective Combination of Genetic Algorithms and the Variable Neighborhood Search for Solving Travelling Salesman Problem, *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 2015, pp. 142-149.
41. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6, 182–197.

42. Solimanpur, M., Vrat, P., and Shanker, R., An Ant Algorithm for the Single Row Layout Problem in Flexible Manufacturing Systems, *Computers & Operations Research*, 32, 2005, pp. 583–598.
43. Anjos, M.F., and Vannelli, A., Computing Globally Optimal Solutions for Single-Row Layout Problems Using Semidefinite Programming and Cutting Planes, *INFORMS Journal on Computing*, 20, 2008, pp. 611–617.