Swarm and Evolutionary Computation 8 (2013) 1–12

Contents lists available at SciVerse ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Regular Paper

Adaptive variable neighborhood search for solving multi-objective facility layout problems with unequal area facilities

Kazi Shah Nawaz Ripon [a,*], Kyrre Glette [a], Kashif Nizam Khan [b], Mats Hovin [a], Jim Torresen [a]

[a] Department of Informatics, University of Oslo, P.O. Box 1080, Blindern, 0316 Oslo, Norway
[b] Computer Science and Engineering Discipline, Khulna University, Bangladesh

ARTICLE INFO

Article history:
Received 30 November 2011
Received in revised form
8 July 2012
Accepted 17 July 2012
Available online 2 August 2012

Keywords:
Multi-objective evolutionary optimization
Variable neighborhood search (VNS)
Adaptive local search
Unequal area facility layout problem
Similarity coefficient measure (SCM)

ABSTRACT

In this paper, we report the results of our investigation of an evolutionary approach for solving the unequal area multi-objective facility layout problem (FLP) using the variable neighborhood search (VNS) with an adaptive scheme that presents the final layouts as a set of Pareto-optimal solutions. The unequal area FLP comprises a class of extremely difficult and widely applicable optimization problems arising in diverse areas and meeting the requirements for real-world applications. The VNS is an explorative local search method whose basic idea is systematic change of neighborhood within a local search. Traditionally, local search is applied to the solutions of each generation of an evolutionary algorithm, and has often been criticized for wasting computation time. To address these issues, the proposed approach is composed of the VNS with a modified 1-*opt* local search, an extended adaptive local search scheme for optimizing multiple objectives, and the multi-objective genetic algorithm (GA). Unlike conventional local search, the proposed adaptive local search scheme automatically determines whether the VNS is used in a GA loop or not. We investigate the performance of the proposed approach in comparison to multi-objective GA-based approaches without local search and augmented with traditional local search. The computational results indicate that the proposed approach with adaptive VNS is more efficient in most of the performance measures and can find near-optimal layouts by optimizing multiple criteria simultaneously.

1. Introduction

The facility layout problem (FLP) is the determination of the most efficient physical arrangement of a number of interacting facilities on the factory floor of a manufacturing system in order to meet one or more objectives. Facilities usually represent the largest and most expensive assets of the organization and are of crucial importance to the organization [1]. A facility is an entity that assists in one dedicated task and can include a department, a machine tool, a work center, a manufacturing cell, a machine shop, or a warehouse [2]. The FLP is a classic computer science problem and has been shown to be NP-hard [3]. Layout planning in a manufacturing company is also an important economical consideration. An effective layout will help any company improve its business performance and can reduce up to 50% of total operating expenses [4]; conversely, an ineffective layout can add as much as 36% to material handling costs [5]. The estimate is that an amount that exceeds 250 billion is spent annually in the United States alone on planning and revising facility layouts [6].

A lot of optimal and heuristic algorithms for solving FLPs have been developed in the past few decades [7–9]. The majority of these approaches adopt a problem formulation known as the quadratic assignment problem (QAP) that is particularly suitable for equal area facilities. The main drawback of these approaches is that geometric constraints, e.g. unequal sizes of facilities, are not taken into account. These approaches tend to focus on the relative location of equal area facilities on a floor plan. If all the facilities are of equal area, or can be physically interchanged without altering the overall adjacency or distance relationship among the remaining facilities, it is easy to specify in advance a finite number of potential sites for these facilities to occupy [10]. However, in most real-world applications, equal area facility is a very poor assumption [11,12].

When layouts have varying area facilities, it can no longer be treated as the problem of assigning n facilities to n distinct centroid locations. Instead, the locations of the centroids will depend on the exact configuration selected, making the QAP formulations of unequal area FLPs less tractable than their equal area counterparts. To handle unequal area FLPs, early heuristic algorithms are based on discrete models which divide the floor plan into a grid of equal-sized squares. Then, each facility is assigned to the number of squares

* Corresponding author. Tel.: +47 22840825; fax: +47 22852401.
E-mail addresses: ksripon@ifi.uio.no (K.S.N. Ripon), kyrrehg@ifi.uio.no (K. Glette), kashif570@yahoo.com (K.N. Khan), matsh@ifi.uio.no (M. Hovin), jimtoer@ifi.uio.no (J. Torresen).

2210-6502/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.swevo.2012.07.003

which most closely matches its total area. While conceptually simple, this approach mostly generates odd-shaped facilities [12]. Some procedures, nonetheless, place restrictions on the shapes of the composite facilities to make the problem more tractable and implementable. The unequal area FLP addresses these situations effectively by finding the optimal arrangement of a given number of non-overlapping indivisible facilities with unequal area requirements on a factory floor. This in turn makes the FLP with unequal areas a fundamental optimization problem encountered in many manufacturing and service organizations.

Most of the studies conducted in FLPs have focused on a single objective, either quantitative (distance-based) or qualitative (adjacency-based) goodness of the layout. In contrast, practical FLPs involve several conflicting objectives. Therefore, both quantitative and qualitative objectives must be considered simultaneously before arriving at any conclusion. A layout that is optimal with respect to a given criterion might be a poor candidate when another criterion is paramount. In general, minimization of the total material handling (MH) cost is often used as the optimization criterion in FLPs. The closeness rating, hazardous movement, safety, and the like are also important criteria in FLPs. In fact, these qualitative factors have significant influence on the final layout and should give consideration. In addition, when only a quantitative objective is used, the criteria are often tempered with judgment to include "inside knowledge" [13]. Consequently, the FLP falls into the category of multi-objective optimization problem (MOOP).

Multi-objective optimization is a technique to treat several objectives simultaneously without converting them into one. The objective of MOOPs is to find a set of Pareto-optimal solutions [14], which are the superior solutions when considering all the objectives. In MOOPs, the absolute optimal solution is absent and the designer must select a solution that offers the most profitable trade-off between the objectives as an alternative. Thus, instead of offering a single solution, it is more realistic and appropriate to generate a number of "good" layouts that meet several criteria laid down by the facility designer and let decision makers choose between them based on the current requirement. Surprisingly, there is a little attention paid to the study of multi-objective FLPs, much less in the case of unequal area FLPs.

Recently, meta-heuristic approaches have been also developed and widely applied to solve large FLPs [8,9]. Among those approaches, the genetic algorithm (GA) has been theoretically and empirically proved to provide a robust search in complex search spaces [9]. Despite of the successful application of GAs to numerous optimization problems including FLPs, it has a major limitation in applying to such problems. GAs can do global search in the entire search space, but there is no way for exploring the search space within the convergence area generated by the GA loop. In certain cases, it performs too slowly to be practical. A very successful way to improve the performance of GAs is to hybridize it with local search. Local search techniques are used to refine the solutions explored by the GA by searching its vicinity for the fittest individuals and replacing it if a better one is found.

The variable neighborhood search (VNS) is a relatively recent meta-heuristic which is based on systematic change of neighborhood within a possibly local search [15,16]. Contrary to other meta-heuristics based on local search methods which use single neighborhood search method only; the VNS explores increasingly distant neighborhoods of the current incumbent solution, and jumps from this solution to a new one if and only if there is an improvement. In this way, it keeps favorable characteristics of the incumbent solution and obtains promising neighboring solutions. By allowing the use of different neighborhood search methods, the VNS can easily escape from local optima and move towards global optimum.

Unfortunately, there has been a little study on the local search technique in multi-objective optimization to date. Then again, hybridization with local search may degrade the global search ability of multi-objective GAs and require more computational time than conventional GAs [17]. This is because, applying local search to all individuals in a population is a computationally intensive procedure. Therefore, most of the computation time is spent in local search. To overcome this weakness, an adaptive local search scheme can be used to automatically control whether the local search is used in a GA loop or not.

Although the advantages and good performance of GAs in FLPs have been demonstrated in the literature, yet there is no formal approach to solve the unequal area FLP considering multiple objectives separately using local search. It is also noticeable that all the existing methods use either single objective or a weighted-sum method to solve unequal area FLPs. As a result, Pareto-optimality was never utilized for solving unequal area FLPs. In our previous paper [11], we proposed a multi-objective GA for unequal area FLPs using Pareto-optimality [11], but without any local search. In this work, we propose a multi-objective GA for solving the unequal area FLP using an adaptive local search scheme to investigate its performances in solving such problems with multiple objectives. The proposed approach presents the layouts as a set of Pareto-optimal solutions. The proposed adaptive local search scheme is based on a modified VNS and the similarity coefficient measure (SCM) to remove the traditional global search problem exhibited by GAs. In addition, we extend the 1-*opt* local search [18] that is integrated within the VNS framework by incorporating domination strategy for handling unequal area multi-objective FLPs. We use the non-dominated sorting genetic algorithm 2 (NSGA 2) [19] as the multi-objective evolutionary algorithm (MOEA).

The paper is organized in the following way. Section 2 presents the related works. Section 3 mentions the importance of Pareto-optimality in solving FLPs. Section 4 justifies the application of the adaptive local search scheme and also describes its implementation. Section 5 outlines the implementation of the proposed unequal area multi-objective FLP approach. This section also includes the layout construction process and the implementation of the adaptive local search using the VNS. Section 6 analyzes the results obtained, followed by the conclusion in Section 7.

2. Related works

Several FLP approaches are available in the literature, which could be mainly classified into (i) exact approaches and (ii) heuristic and meta-heuristic approaches. The biggest limitation of exact approaches is that they cannot optimally solve large FLPs due to the computational intractability of the problem. Optimal algorithms have been successfully applied to small FLPs, but they require high computational efforts and extensive memory capabilities. A recent survey shows that the results achieved by the best existing exact algorithms (Branch and Bound) are modest. It is not suitable for solving FLPs of size larger than 20 facilities in reasonable time [20]. Thus, researchers have relied on heuristic methods for searching through the huge search space which is representative of practical FLPs.

Armour and Buffa [21] presented the first formulation for the unequal area FLP with pair-wise exchange method. Since then, quite a few authors have attempted to address unequal area FLPs. The recent reviews of unequal area FLPs are given in [1,7]. Konak et al. [22] developed a mixed integer programming (MIP) formulation to solve unequal area FLPs based on the flexible bay structure (FBS), which can find optimal solutions for problems with up to 14-facilities only. In terms of meta-heuristics, Castillo et al [23] applied a mixed-integer nonlinear programming for solving this problem. Hu and Wang [24] applied GA to unequal area FLPs for achieving the minimal layout cost. Tate and Smith [10] presented a GA-based model for FLPs with unequal areas and different geometric shape constraints. Unfortunately, all of these approaches are for optimizing a single objective. In [1], a hybridized meta-heuristic for solving the unequal area FLP is presented. However, it is mainly based on discrete representation. Recently, a GA- and MIP-based approach [25], a tabu search (TS)-based approach with slicing tree representation [26], an ant system (AS)-based method [27], and an ant colony optimization (ACO) combined with FBS-based approach [28] have been proposed to solve unequal area FLPs. Similar to the GA-based approaches, all of these approaches focus on single objective. A multi-objective approach for solving the unequal area FLP has been proposed in [29]. However, it used the weighted-sum method to handle multiple objectives.

3. Importance of Pareto-optimality in the FLP

The typical approaches to FLPs have been implemented to optimize only one objective: either quantitative or qualitative objective. Quantitative objective aims at minimizing the sum of the product of material flow, distance, and transportation cost per unit per distance unit for each pair of facilities. It is popularly known as the material handling (MH) cost. Qualitative objective aims to place facilities that utilize common materials, personnel, or utilities adjacent to one another, while separating facilities for the reasons of safety, noise, or cleanliness. Qualitative objective uses a relationship chart to maximize the overall adjacency measure for a given layout. This chart specifies the closeness rating (CR) for each facility pair.

Many researchers have questioned the appropriateness of selecting a single criterion to solve FLPs because qualitative and quantitative approaches each have advantages and disadvantages [30]. The major limitations on quantitative approaches are that they consider only relationships that can be quantified and do not consider any qualitative factors. On the other hand, qualitative approaches suffer from the shortcoming of strong assumption made on all qualitative factors that these factors can be aggregated into one criterion. Moreover, in MOOPs, obtaining an “absolute optimum” solution that satisfies all objectives is almost impossible. It is due to the conflicting nature of objectives, where improving one objective may only be achieved when worsening another objective. In such case, it is desirable to generate a set of approximately efficient solutions—the Pareto-optimal solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior considering all the objectives. Therefore, the decision maker can pick up the best solution among all of the generated solutions for specific order or customer demands.

Although researchers have been proposing approaches for solving the multi-objective FLP over the last few years [7, 9], these approaches, in most cases, lead to the optimization of a weighted-sum of a function. In this method, multiple objectives are added up into a single, scalar objective using weighted coefficients. The impractical weighted-sum approach involves the difficulty of normalizing these objectives and of quantifying the weights in advance. In addition, other disadvantages of this technique include issues with predetermining the relative weights of objectives, obtaining inferior non-dominated solutions, user involvement in specifying the weight values, and obtaining a single solution at one time. Interested readers can find the details in [30, 31]. To overcome these weaknesses, Pareto-optimality has become an efficient alternative.

4. Rationale for the adaptive local search

Although GAs can find promising regions quickly while solving combinatorial optimization problems, it can suffer from excessively slow and premature convergence before providing an accurate solution. Also, GAs have inherent difficulties in converging to the global optimum with an adequate precision in complex and large search space that is very usual in real-world FLPs. This is because of GA’s fundamental characteristics—not using a priori knowledge and inability to explore the search space within the convergence area generated by the GA loop. In contrast, local search heuristic can iteratively examine a set of points in the neighborhood of the current solution and replaces it if a better neighbor exists. Therefore, the synergy between both methods can give rise to a family of hybrid algorithms, simultaneously global and precise. The GA globally explores the domain and finds a good set of initial estimates, while the local search further refines these solutions in order to locate the nearest, best solution.

Hybridization with local search, however, often degrades the global search ability of the GA. This is because, local search is usually applied to the solutions of each generation of the GA. As a result, the local search technique has to examine a large number of solutions for finding a locally optimum solution from each initial solution generated by genetic operations. It is nothing but mere waste of CPU time. Also, most of the local search techniques used in hybridized GAs are designed without any analysis of their convergence characteristics [32]. For decreasing the computation time spent by local search and improving these weaknesses in the application of local search, we implemented an adaptive local search technique augmented with the VNS that will only search around the convergence area produced by the GA loop instead of applying to all individuals.

The basic idea behind the proposed adaptive local search scheme is to consider whether the GA is converging to global optimal solution or not. When the GA is converging to a global optimum, the solutions are continuously improved. In this situation, the application of local search is needless in order to save the unnecessary use of computation time. On the contrary, the performance of a GA definitely deteriorates if it is not converging to global optimal solution. At the worst, if this situation continuously proceeds, it will get stuck into a local optima resulting in a premature convergence. A local search technique within the GA loop helps improving this situation, since it can generate new individuals having certain high fitness values like the superior individuals generated by the GA.

5. Implementation

5.1. Chromosome representation

The proposed approach uses the slicing tree structure [27] for representing chromosomes suitable for unequal area FLPs. slicing tree is a binary tree which is used to represent a slicing structure. For a FLP with n acilities, the slicing tree consists of n leaves and $n-1$ internal nodes, where each leaf represents a facility, and each internal node contains information about the direction of cut (horizontal or vertical).

In this work, the chromosomes are divided into three parts and are encoded as $(f_1 f_2 f_3, \ldots, f_n)$ $(ss_1 ss_2 ss_3, \ldots, ss_{n-1})$ $(so_1 so_2 so_3, \ldots, so_{n-1})$, respectively, where f, ss, so, and n represent facility sequence, slicing sequence, slicing orientation, and the number of facilities, respectively. The first two parts of the chromosome are represented by integers, whereas the last part is represented by either 1 or 0. The facility sequence will be transformed into a slicing tree form. The slicing sequence is the ordering that slices the facility sequence. A slicing orientation 0 represents a horizontal cut and 1 represents a vertical cut. A chromosome for a 7-facility problem is shown in Fig. 1. Every slicing structure can be represented by a slicing tree and vice versa, but there can be multiple slicing trees corresponding to the same slicing structure. In general, the slicing tree representation recursively divides the total floor area either in horizontal or vertical direction completely from one side to the other in proportion to the areas of the facilities. Fig. 2 presents the corresponding solution representation, the slicing tree transformation, and the layout for the chromosome presented in Fig. 1.

5.2. Objective function

In this work, we follow the assumptions described in [27]: facilities must be located within a given area; facilities must not overlap with each other; the layout must fulfill the maximum aspect ratio constraints (or minimum value restrictions) for the dimension of facilities. Aspect ratio constraints are frequently used in FLPs to restrict the occurrence of overly long and narrow facilities in the layout, and are measured as the ratio between the height and width of a facility.

We use the total MH cost as the first objective which is based on quantitative model. The second objective, the CR score, is based on qualitative model. The first objective is subject to minimization, while the later one is subject to maximization. These objectives can be expressed by the following mathematical models:

$$F_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij} f_{ij} d_{ij} \tag{1}$$

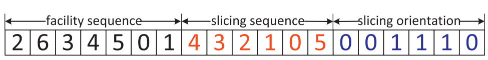

Fig. 1. Chromosome representation for a 7-facility problem.

$$F_2 = \sum_{i=1}^{n}\sum_{j=1}^{n} r_{ij} \tag{2}$$

subject to

$$r_{ij} = \begin{cases} \text{CR value} & \text{if facility } i \text{ and } j \text{ are neighbors with a common boundary} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where f_{ij}, C_{ij} is the material flow, and the transportation cost between facilities i and j, d_{ij} is the Euclidean distance between the centers of facilities i and j. We use the following CR values: absolutely necessary = 6, essentially important = 5, important = 4, ordinary = 3, un-important = 2, and undesirable = 1.

5.3. Layout construction

In this approach, we assume that the total facility area cannot be larger than the sum of width and height of all facilities. This is obvious because in the worst case scenarios, all the facilities might be placed horizontally or vertically. In such cases, the final width and height of the layout will be equal to the sum of widths or the sum of heights of all facilities. In other cases, the final width and height of the layout will be less than the sum of widths and heights of all facilities.

At the beginning when no facilities are assigned to any specific location, the upper left co-ordinate ($x1$, $y1$) of all facilities are (0,0) and the lower right co-ordinate ($x2$, $y2$) will be ($\sum$ widths of all facilities, $\sum$ heights of all facilities). It indicates that initially there is

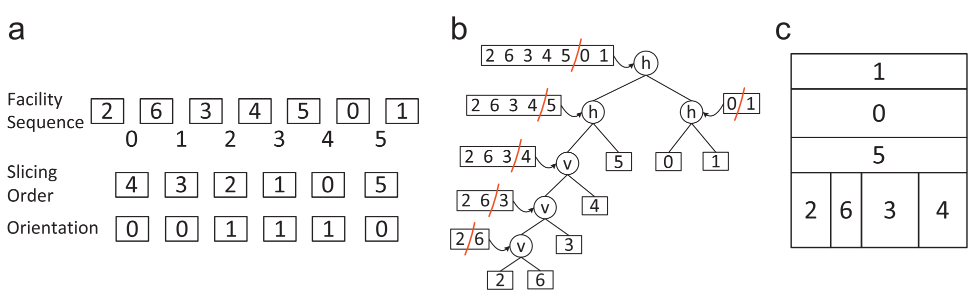

Fig. 2. Transformation of chromosome into slicing tree and the corresponding layout. (a) Solution representation, (b) slicing tree and (c) solution layout.

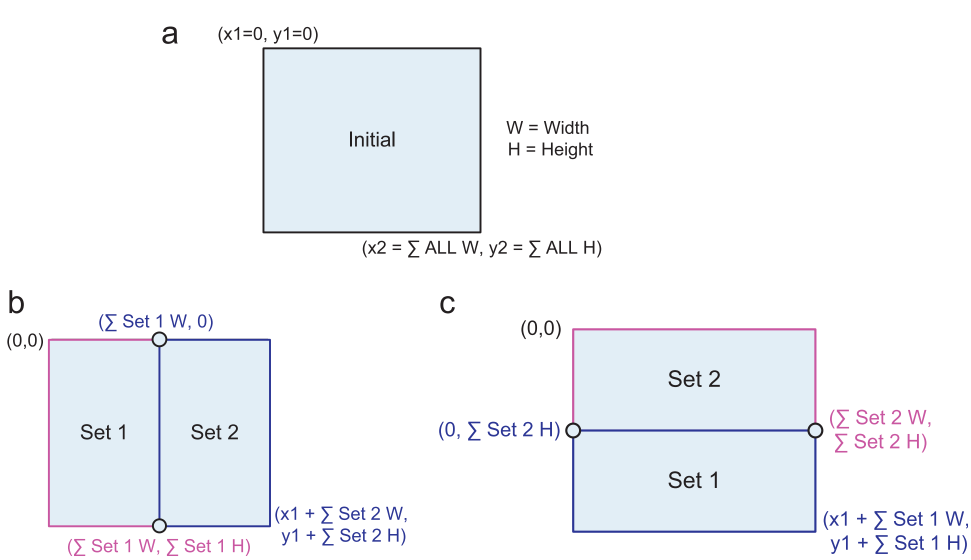

Fig. 3. Placement of facilities (first step). (a) Initial layout area. (b) Vertical cut (orientation type = 1). (c) Horizontal cut (orientation type = 0).

one complete layout area which holds all facilities. Still the specific positions of the facilities within that area is unknown. At this point, we start assigning specific positions to each facility according to the slicing sequence and orientation. The slicing sequences are processed one at a time. Each slice breaks the layout area into two pieces. The co-ordinates of the facilities (or facility) for the two newly created smaller areas are calculated according to the orientation of the slice. (Fig. 3(b)) describes the first step of facility placement.

If the first slice has a vertical cut, all facilities on the left side of the cut (in the original chromosome) will be in Set 1 and those that are on the right side will be in Set 2 (Fig. 3(b)). The ($x1$,$y1$) of Set 1 will remain as (0,0). However, the ($x2$,$y2$) will be recalculated as ($\sum$ widths of all facilities in Set 1, $\sum$ heights of all facilities in Set 1). Since Set 2 does not start from the point (0,0) after slicing, the ($x1$,$y1$) and ($x2$,$y2$) of Set 2 will be recalculated as follows:

$$(x1,y1) \text{ of Set } 2 = \left(\sum \text{widths of all facilities in Set } 1, 0\right)$$

$$(x2,y2) \text{ of Set } 2 = \left(x1 + \sum \text{widths of all facilities in Set } 2, y1 + \sum \text{ heights of all facilities in Set } 2\right)$$

These calculations can be easily derived from Fig. 3(b). On the other hand, if the first slice has a horizontal cut; all facilities on the left side of the cut (in the original chromosome) will be in the lower portion of the cut which is Set 1. And, the facilities which are on the right side will be in higher portion of the cut which is Set 2. Similar to Fig. 3(b), the ($x1$,$y1$) and ($x2$,$y2$) for the newly created set of facilities have to be recalculated. This is depicted in Fig. 3(c). This recalculation procedure is performed recursively for every newly created set of facilities until all the facilities have been placed and all the sets contain only one facility.

Fig. 4 gives a pictorial description of the facility placement procedure for the 7-facilty problem shown in Fig. 2. In this figure, the height and width ratios of the facilities do not follow the actual scale measure. The (area, maximum aspect ratio) pairs for the problem are {(16, 4), (16, 4), (16, 4), (36, 4), (9, 4), (9, 4), (9, 4)}. After a certain point, this figure does not include ($x1$, $y1$),($x2$, $y2$) for clarity and better understanding. The pseudo-code for the procedure is given in Fig. 5.

5.4. Variable neighborhood search (VNS)

The VNS is closely related to iterated local search (ILS). However, instead of iterating over one constant type of neighborhood structure as done in ILS, the VNS switches neighborhoods of growing size to identify better local optima with shaking strategies [33]. The steps of basic VNS are presented in Algorithm 1, where N_k represents the kth neighborhood structure ($k = 1, \ldots, k_{max}$), S represents the set of all feasible solutions, and $N_k(s)$ represnts the set of all solutions in the kth neighborhood of the solution s.

```
void recursive_positioning (int start, int end, int slice_position)
if slice_position is within start and end && there are more than one element in the set then
  create two sets: Set 1 from start to slice_position and Set 2 from slice position to end
  calculate w1 ← ∑ Set 1 widths, w2 ← ∑ Set 2 widths, h1 ← ∑ Set 1 heights, h2 ← ∑ Set 2 heights
  if orientation == horizontal then
    for Set 2 do
      Set 2 . y2 ← Set 2 . y1 + h2
      Set 2 . x2 ← Set 2 . x1 + w2
    end for
    for Set 1 do
      Set 1 . y1 ← Set 1 . y1 + h2
      Set 1 . y2 ← Set 1 . y1 + h1
      Set 1 . x2 ← Set 1 . x1 + w1
    end for
  else
    for Set 1 do
      Set 1 . y2 ← Set 1 . y1 + h1
      Set 1 . x2 ← Set 1 . x1 + w1
    end for
    for Set 2 do
      Set 2 . x1 ← Set 2 . x1 + w1
      Set 2 . y2 ← Set 2 . y1 + h2
      Set 2 . x2 ← Set 2 . x1 + w2
    end for
  end if
  find the next_slice_position
  recursive_positioning (start, slice_position , next_slice_position)
  recursive_positioning (slice_position, end, next_slice_position)
end if
```

Fig. 5. Pseudo-code for the facility placement procedure.

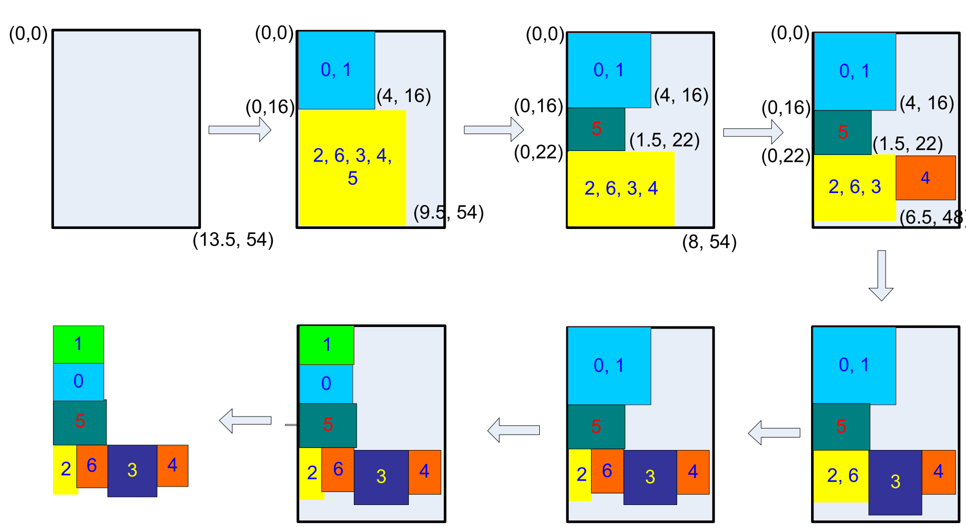

Fig. 4. Facility placement procedure.

Algorithm 1. Basic VNS Algorithm.

Initialization:
Define neighborhood structures N_k, $k=1,\dots,k_{max}$;
Find an initial solution $s \in S$;
Choose an end condition;
while end condition is not met **do**
  Set $k=1$;
  while $k \le k_{max}$ **do**
    Shaking:
    Generate a point $s' \in N_k(s)$ at random;
    Local search:
    Obtain the local optimum s'' by applying some local search to s';
    Move or not:
    if s'' is better than s **then**
      $s=s''$;
      $k=1$;
    else
      $k=k+1$;
    end if
  end while
end while

As stated earlier, changing of neighborhood within a possibly local search is the basic principle in the VNS. So how to define the effective neighborhood around an initial solution is crucial in the VNS. Many researchers have concluded that the insertion neighborhood is superior to the swap or exchange neighborhood in the case of permutation representation [34], which we use in this work.

In this approach, two different types of insertion neighborhood structure each comprising of two different orders are adopted. These two types are divided based on the number of chromosome segment taking part in constructing the neighborhood. For simplicity, we consider only the first two segments (facility sequence and slicing sequence) for this process. First, we randomly decide the number of segments. If it is one, we choose the segment at random. One gene from this segment is randomly removed from its position and inserted elsewhere within the same segment. All genes within these positions are moved one position forward or backward. We perform the same process for both of the segments, if both segments are selected for the process. For the other two neighborhoods, we repeat the same process (single segment or double segments) but changing the number of genes. Instead of one gene, two genes are removed from their positions and inserted elsewhere within the same segment. Therefore, the VNS interactively explores neighborhood of growing size.

In the *local search* step of the VNS (Algorithm 1), we use the modified 1-*Opt* local search method proposed in Section 5.4.1. Since, we are dealing with multiple objectives, we have to modify the next step (*Move or not*) of the VNS algorithm. Here "better" means non-dominated. A neighbor is made the current solution if and only if it dominates the current one.

5.4.1. Modified 1-Opt Local Search

In order to apply local search, we have to specify an objective function to be optimized by the search. This specification is straightforward for a single objective optimization problem because the single objective function can be used for both genetic and local search. As mentioned earlier, there are much fewer studies of hybridization for multi-objective optimization. A weighed-sum of multiple objectives is often used for local search in hybrid multi-objective GA [17], which is not realistic.

Algorithm 2. Modified 1–*Opt* Local Search for Unequal Area MOFLP.

set $fit_1^1 =$ MH cost of the current chromosome;
set $fit_2^1 =$ CR score of the current chromosome;
for $i=1$ to $2n-1$ **do**
  repeat
    select another gene j randomly such that
    (i) $i \ne j$; AND
    (ii) i, j will be from the same portion (facility sequence or slicing sequence);
    construct a new layout by swapping i and j;
    find $fit_1^2 =$ MH cost of the new layout;
    find $fit_2^2 =$ CR score of the new layout;
    if the current layout is dominated by the new layout **then**
      replace the current one with the new layout;
    else
      continue with the current one;
    end if
    until any improvement in the current layout
end for

In this work, we extend the 1-*opt* local search [18] for the unequal area multi-objective FLP by incorporating domination strategy [19]. The general outline of this algorithm is given in Algorithm 2. This implementation of local search is to replace the current solution with its neighbor that dominates the current solution. The incorporation of domination strategy is necessary because we are dealing with multiple objectives and all the previous applications of 1-*opt* local search were proposed for single objective only. By employing this strategy, we can find out which of the neighbors of the current solution dominates the current one (if any exists). Also, to save computation time and effort, this algorithm stops searching the neighborhood as soon as it finds a solution which dominates the current one.

5.5. Implementation of adaptive local search

The proposed adaptive local search scheme is based on the VNS and the SCM to consider the similarity of individuals of a GA population. The basic idea of the SCM was originally developed for grouping machines and parts in production environment [35]. When the GA is continuously converging, the similarity among the individuals of a GA population becomes higher. Therefore, the fitness values of the individuals are significantly similar to each other and the variety of the population is reduced. This definitely deteriorates the performance of the GA [32]. Inserting new individuals into the current population can improve these situations. These new individuals should not have too close similarity with the current population and also preferably get high fitness values compared to the rest of the population.

In this work, we modified the basic idea of SCM for the unequal area multi-objective FLP environment. We can calculate the similarity coefficient SC_{pq} between two chromosomes p and q as follows:

$$SC_{pq} = \frac{\sum_{k=1}^{n} \delta(f_{pk}, f_{qk})}{n} \tag{4}$$

where f_{pk} and f_{qk} are the facilities at location k in chromosomes p and q, k is the index of location in the layout (only within the facility sequence portion of the chromosome), and n is the number of facilities.The genes of the first and the second portion of chromosome contain any values within the range of 1 to n, and 1 to $n-1$, respectively. Whereas, the third portion contains only 1 or 0. So, to compare the similarity between two chromosomes, $\delta(f_{pk}, f_{qk})$, we only consider the first two portions. This can be expressed as follows:

$$\delta(f_{pk},f_{qk}) = \begin{cases} 1 & \text{if } (f_{pk} == f_{qk}) \text{ AND } (ss_{pk} == ss_{qk}) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where ss_{pk} and ss_{qk} are the slicing sequences at location k in chromosomes p and q, and k is the index of slicing sequence portion of the chromosome in the layout (in this portion, the maximum value of k will be $n-1$). The average similarity coefficient ($\overline{SC}$) for all individuals of the population (N) can be expressed as follows:

$$\overline{SC} = \frac{\sum_{p=1}^{N-1} \sum_{q=p+1}^{N} SC_{pq}}{N} \quad (6)$$

Assuming a pre-defined threshold value (β), the local search method in form of the VNS will be automatically invoked in a GA loop by the following condition:

$$\begin{cases} \text{apply VNS to GA loop} & \text{if } \overline{SC} > \beta \\ \text{apply GA alone} & \text{otherwise} \end{cases} \quad (7)$$

5.6. Crossover

We applied 3–point crossover operation depicted in Fig. 6. For keeping the chromosomes valid after the exchange of genes, we choose the 3 points separately from each segment of a chromosome. However, for the first two segments, some repair operations are required after the exchange to remove any duplication or absence of genes. For the repair operations, first we find and list the duplicate facilities in the first segment according to the occurrence in the chromosome. Then, we check whether any facility is missing in the segment starting from the first to the last facility (from 1 to n). After that we replace the list of the duplicate facilities with facilities that are missing. The same procedure is repeated for the genes of the second segment, except that here the range is from 1 to $n-1$.

5.7. Mutation

We use swap mutation with the restriction that both genes will be chosen from the same segment. As a result, no repair is necessary. Unlike the crossover, the genes will be chosen from only one segment of the chromosome and this choice will be random for every chromosome of the population pool. Fig. 7 gives an example for the mutation.

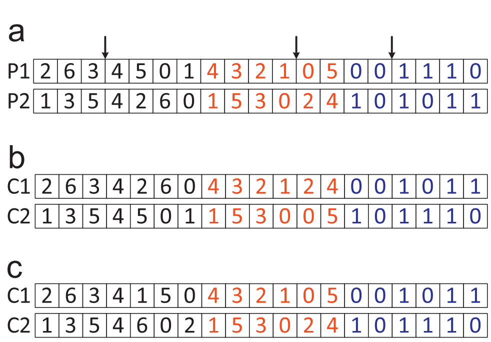

Fig. 6. Crossover operation. (a) Parent chromosomes before crossover. (b) Child chromosomes after crossover without repair. (c) Child chromosome after repair.

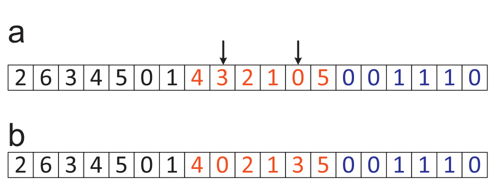

Fig. 7. Mutation. (a) Before. (b) After.

6. Computational result and analysis

6.1. Test setup

To evaluate the efficiency of the proposed approach, the results are compared to the unequal area multi-objective FLP approach without any local search [11]. To our knowledge, it is the only available published results for the unequal area multi-objective FLP using Pareto-optimality. We also implement an approach for solving the unequal area multi-objective FLP with traditional local search, and compare with our proposed adaptive VNS-based local search approach. This will give an idea about the performance of the proposed adaptive VNS scheme in comparison to the traditional local search.

The experiments are conducted using various benchmark instances taken from published literature. The test problems are composed of 7, 8, 9, 10, 12, 14, 20, 30, 35, and 62 facilities. The details of the problem data can be found in [27]. It is worthwhile to mention that very few benchmark problems are available for unequal area multi-objective FLPs, particularly in the case of CR score. As a result, we have created CR scores for these data sets on our own. We use the last digits to indicate the number of facilities in each problem. We set the maximum aspect ratios as 4 for *Ba*12, *Ba*14, *SC*30, and *SC*35 which are not fixed in the original literature. To justify the proposed approach, we also compare with some existing single objective (MH cost) heuristic approaches for unequal area FLPs.

The experiments are conducted using 200 chromosomes and 100 generations for problems with up to 15 facilities; and 1000 chromosomes and 900 generations for problems with more than 15 facilities. However, for justifying the convergence behavior of the proposed adaptive local search, later we run the three multi-objective unequal area FLP approaches less than the mentioned generations. The probabilities of crossover, mutation and pre-defined coefficient (β) are 0.9, 0.3, and 0.9 (90%), respectively. We use the traditional tournament selection with a tournament size of 2. The stopping condition for the VNS is set to 5% of the number of chromosomes. Each benchmark problem is tested for 30 times with different seeds. Then each of the final generation is combined and a non-dominated sorting is performed to constitute the final non-dominated solutions.

6.2. Experimental analysis

Since almost all FLP approaches try to optimize single criteria only (mainly minimizing the MH cost), first we compare the MH costs obtained by our approach with the existing single objective approaches. Then we show its performance as an unequal area multi-objective FLP approach by optimizing both MH cost and CR score. Note that, for both single and multiple objectives, we use the same results from the same non-dominated solutions obtained by the approaches.

In single-objective optimization, "quality" can be easily defined by means of the objective function—the smaller (or larger) the value, the better the solution. However, when dealing with MOOPs, there are several reasons why a qualitative assessment of results becomes difficult [36, 37]. The initial problem is that MOOPs do not try to find one optimal solution but all the trade-off solutions. There are two distinct goals in MOOPs: (i) convergence to the true Pareto-optimal front, and (ii) diversity of the Pareto-optimal solutions. Also, due to the stochastic nature of EAs, multiple runs on the same problem are necessary to get a good estimation of performance [38]. Accordingly, it is not clear what "quality" means with respect to Pareto front in MOOPs. Thus, the quality of multi-objective optimization is often difficult to define precisely by any single performance metric and the results have to be validated using statistical analysis tools. Recently, a few studies have been carried out to clarify this situation [36, 38, 39]. Generally, researchers utilize various methods that assign a vector of real numbers to each Pareto front that reflect different aspects of quality. These elements of the vector are called the unary quality indicators. A list of unary indicators that have been introduced over the past few decades is available in [36].

To analyze the performance of the final layouts optimizing multiple objectives, we use the following three metrics (unary indicators) mentioned in [36]. We choose these matrices because they do not require the prior knowledge of the true Pareto front which is not available for the test problems.

1. *Pareto Ratio* (*PR*) [40]: It is related to the convergence measurement of a Pareto front. It varies from 0 to 1 and the ideal value of a population is 1.
2. *Space* (*S*) [41]: The metric *S* is used to measure the diversity of the non-dominated solutions. As long as the spread is uniform within the range of solutions obtained, this metric produces a small value.
3. *Overall Pareto Spread* (*OPS*) [42]: This metric is also related to the diversity. *OPS* quantifies how widely the non-dominated solutions spreads over the objective space considering all the objectives. When comparing two Pareto fronts, the one with wider spread is desirable. In other words, if $OPS(PF_1) > OPS(PF_2)$, the PF_1 is preferred to PF_2.

The values provided in Table 1 show the MH cost for the best layouts obtained by our approach and some existing algorithms. We compare our results with those obtained by ant system (AS) [27], GA with flexible bay representation [10], GA with MIP [25], TS with slicing tree representation [26], and our previous approach without any local search [11]. This table is partially cited from [11, 23]. The best results are bold-faced in the table. As shown in the table, both the proposed approaches (with adaptive VNS and with traditional local search) outperform AS and GA with flexible bay for all the test problems. They perform better than TS with slicing tree for all the problems except *Ba*14, where they find the same MH cost. In comparison to GA with MIP, the proposed adaptive VNS-based approach finds better results for all the problems except *SC*35, and the result is the same in *SC*30.

It is interesting to observe that the performances are significantly improved with the introduction of local search. In fact, both the approaches with local search (adaptive and traditional) find new best results for *Ba*12 and the same as the existing best result for *Ba*14. Most importantly, the incorporation of adaptive VNS scheme helps the algorithm to achieve the new best solutions for *VC*10, *Ab*20, *SC*30, and *Du*62. It is also noticeable that the approach with traditional local search minimizes its MH cost in comparison to our previous approach without any local search. All these justify the application of adaptive VNS scheme in solving unequal area FLPs. Above all, the values in the table suggest that our approach with adaptive VNS performs well in cases of both small and large FLPs.

Table 2 compares the results of the proposed adaptive VNS-based multi-objective FLP approach with conventional local search and without local search [11] in the context of MH cost and CR scores. The results shown in the table indicate that the adaptive VNS-based approach clearly outperforms the others. Indeed, it achieves better MH cost for all the problems except 07, 08, and 09 in comparison to the approach without local search. For these problems, the total MH costs are the same. Then again, while comparing to the approach with traditional local search, it finds better MH costs for *VC*10, *Ab*20, *SC*30, *SC*35, and *Du*62. In the case of CR scores, the adaptive scheme finds better results than the approach without local search for all the problems except 07. For this problem, the CR scores are the same. In comparison to the traditional local search-based approach, our proposed adaptive VNS scheme finds better results for 7 out of 10 problems. The results are the same for the rest three problems. From the table, we can find that the adaptive VNS-based approach is never outperformed by any of the comparing approaches for any of the two objectives. Furthermore, the average values for both objectives considerably improve with the introduction of the adaptive scheme.

This can be further justified by Fig. 8, where the convergence behavior of the proposed and previous methods over generations for both objectives is depicted. From the figures, it can be found that from first generations to last generations, the proposed approaches are able to optimize both MH cost (minimize) and CR score (maximize) successfully. However, the incorporation of adaptive VNS reduces the gaps between the best and average values more than that of the competing approaches for both the objectives. The best and average values obtained by the approaches as mentioned in Table 2 also justify this.

In conjunction with the averages, box plot is an effective way to describe the spread of data. Fig. 9 presents box plots for both objectives to show the distribution tendency of the final Pareto-optimal layouts. Along with the median, the mean is also presented in the boxes as the small square with a cross. From the plots, it can be easily summarized that the proposed VNS-based approach successfully finds the optimal values for both the objectives. The median values for MH cost (minimize) and CR score (maximize) obtained by the approach with adaptive local search are better than the two competitors. Also, the inter-quartile range (second-order moments for the spread of data) of the adaptive scheme is larger than those of the other two. Considering these all, the performance of the adaptive VNS-based approach is consistent enough.

Table 3 highlights the values of *PR*, *S*, and *OPS* metrics for the approaches. As discussed earlier, these metrics are used to illustrate the convergence and diversity of Pareto-optimal layouts for handling multiple objectives. The values of the metrics as shown in this table indicate that the Pareto-optimal layouts obtained by the adaptive VNS scheme have better convergence and diversity characteristics

Table 1
Comparison with existing algorithms for MH cost only.

Data sets	AS [27]	GA with flexible bay [10]	GA with MIP [25]	TS with slicing tree [26]	Without LS [11]	Traditional LS	Adaptive VNS
07	131.68	NA	131.63	132.00	**98.69**	**98.69**	**98.69**
08	243.12	NA	245.41	243.16	**202.72**	**202.72**	**202.72**
09	236.12	NA	246.26	239.07	**201.75**	**201.75**	**201.75**
*VC*10	19 967.60	23 671.00	19 997.00	19 994.10	19 963.74	19 907.75	**19 901.32**
*Ba*12	8252.67	8768.00	8702.00	8264.00	8103.85	**7982.24**	**7982.24**
*Ba*14	4724.68	5080.00	4852.00	**4712.33**	4790.83	**4712.33**	**4712.33**
*Ab*20	4972.56	NA	5668.00	5225.96	4015.25	3982.54	**3978.24**
*SC*30	3868.54	5743.00	**3707.00**	NA	3740.75	3716.44	**3707.00**
*SC*35	4132.37	NA	**3604.00**	NA	3835.78	3638.20	3636.75
*Du*62	3 720 521.13	NA	NA	NA	2 977 512.96	2 976 120.11	**2 975 604.25**

Table 2
Comparison with exiting MOFLP approaches.

Data sets	Comparing approaches	MH cost		CR score		Optimal sol. after 60% generations	Time (s)
		Best	Avg	Best	Avg		
07	Adaptive VNS	98.69	102.36	54	33.125	83%	3.18
	Traditional LS	98.69	102.36	54	33.125	43%	3.18
	Without LS	98.69	111.009	54	30.825	16%	3.18
08	Adaptive VNS	202.72	270.125	70	42.328	76%	4.09
	Traditional LS	202.72	279.69	68	38.00	21%	4.09
	Without LS	202.72	292.602	64	32.267	7%	4.02
09	Adaptive VNS	201.75	294.36	96	68.16	82%	9.46
	Traditional LS	201.75	304.12	94	56.52	18%	9.68
	Without LS	201.75	444.672	88	39.897	15%	9.12
VC10	Adaptive VNS	19 901.32	24 783.75	150	81.016	72%	14.29
	Traditional LS	19 907.75	25 739.038	132	76.048	16%	14.92
	Without LS	19 963.74	26 065.55	120	61.846	0%	11.72
Ba12	Adaptive VNS	7982.24	8591.18	128	78.09	68%	27.42
	Traditional LS	7982.24	8652.94	128	77.62	11%	29.07
	Without LS	8103.85	8711.847	110	71.25	0%	24.27
Ba14	Adaptive VNS	4712.33	5397.04	198	105.96	70%	46.23
	Traditional LS	4712.33	5410.04	184	101.06	4%	51.91
	Without LS	4790.83	5500.217	156	98.167	0%	32.18
Ab20	Adaptive VNS	3978.24	5856.25	216	108.86	57%	367.2
	Traditional LS	3982.54	5892.946	210	104.84	0%	469.02
	Without LS	4015.25	6131.369	201	99.329	0%	296.56
SC30	Adaptive VNS	3707.00	4146.27	380	264.62	58%	732.84
	Traditional LS	3716.44	4186.04	370	249.834	0%	919.01
	Without LS	3740.75	4444.037	349	241.872	0%	587.78
SC35	Adaptive VNS	3636.75	4190.25	340	237.65	52%	1086.72
	Traditional LS	3638.20	4214.075	340	226.96	0%	1422.16
	Without LS	3835.78	4305.899	337	216.043	0%	816.74
Du62	Adaptive VNS	2 975 604.25	3 218 854.65	306	201.125	56%	2847.35
	Traditional LS	2 976 120.11	3 220 214.64	284	168.17	0%	3448.29
	Without LS	2 977 512.96	3 220 771.01	248	155.25	0%	2258.67

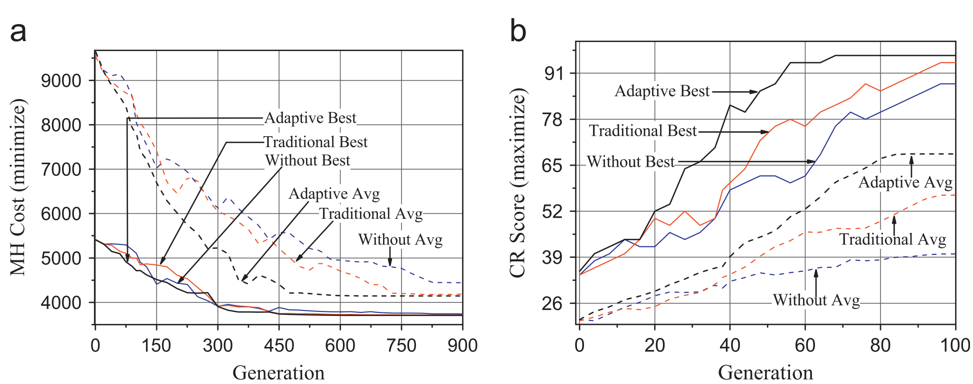

Fig. 8. Two objectives over generations. (a) MH cost (for *SC*30). (b) CR score (for 09).

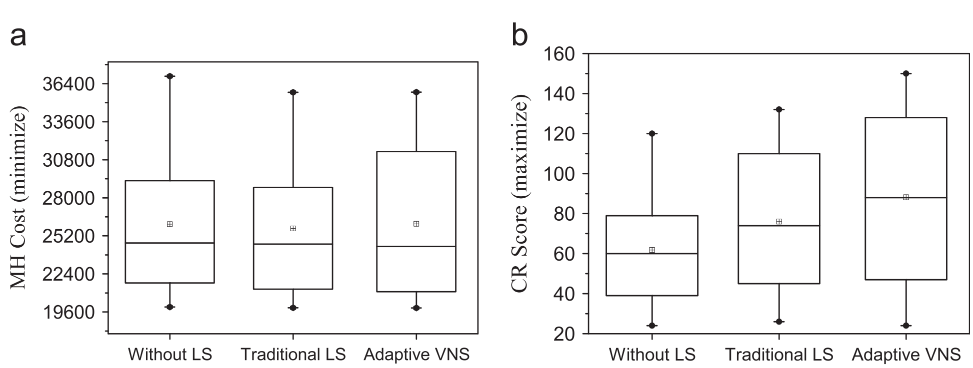

Fig. 9. Box plots for *VC*10. (a) MH cost. (b) CR score.

than the approaches with traditional local search and without local search. Interestingly, the approaches with adaptive VNS and traditional local search have almost the same *PR* and *S* values for small FLPs. However, as the problem size increases, the approach with adaptive scheme shows better behavior than the approach with traditional local search. This also demonstrates the importance of adaptive VNS in obtaining stable solutions.

Since we look for maximizing the total MH costs and minimizing the CR scores, we set the "good point" as (lowest MH cost -10%, highest CR scores$+10\%$) and "bad point" as (highest MH cost$+10\%$, lowest CR scores -10%) for calculating *OPS*. If we observe the values of *OPS* achieved by the three multi-objective approaches, we can find that our proposed approach with adaptive VNS finds the larger *OPS* value for most of the test problems. This justify that our proposed VNS-based approach is capable of finding non-dominated solutions with wider spread. Thus, it can provide a wide range of alternative layouts to choose.

Table 3
Performance metrics for the comparing approaches.

Data sets	Approach	PR	S	OPS
07	Adaptive VNS	1	0.1	0.81
	Traditional LS	1	0.1	0.81
	Without LS	0.98	2.431	0.78
08	Adaptive VNS	0.86	1.416	0.64
	Traditional LS	0.78	2.586	0.60
	Without LS	0.58	5.850	0.64
09	Adaptive VNS	0.86	3.014	0.60
	Traditional LS	0.77	3.054	0.61
	Without LS	0.64	4.721	0.59
*VC*10	Adaptive VNS	0.77	1.049	0.61
	Traditional LS	0.62	1.390	0.51
	Without LS	0.62	1.700	0.50
*Ba*12	Adaptive VNS	0.82	1.668	0.59
	Traditional LS	0.805	1.654	0.60
	Without LS	0.7	2.690	0.59
*Ba*14	Adaptive VNS	0.76	1.974	0.60
	Traditional LS	0.645	2.040	0.60
	Without LS	0.505	2.701	0.56
*Ab*20	Adaptive VNS	0.70	5.892	0.66
	Traditional LS	0.61	6.942	0.62
	Without LS	0.505	7.370	0.60
*SC*30	Adaptive VNS	0.61	5.001	0.67
	Traditional LS	0.51	5.361	0.62
	Without LS	0.49	6.971	0.54
*SC*35	Adaptive VNS	0.58	6.035	0.63
	Traditional LS	0.505	6.473	0.59
	Without LS	0.49	8.374	0.54
*Du*62	Adaptive VNS	0.61	4.952	0.62
	Traditional LS	0.57	5.587	0.56
	Without LS	0.47	7.430	0.53

To further demonstrate the convergence and diversity behavior of the final Pareto-optimal layouts, non-dominated solutions of the final generation produced by the approaches for 08 and *VC*10 are shown in Fig. 10. In fact, many of the final solutions are Pareto-optimal. In the figures, the occurrences of the same non-dominated solutions are plotted only once. The value of *PR* metric for each problem mentioned in Table 3 also indicate this phenomenon. From these figures, it can be observed that the final solutions for all approaches, particularly with adaptive VNS scheme, are well spread and converged. It can be further justified by the values of *S*, *PR*, and *OPS* as specified in Table 3. And for this reason, the proposed approach is capable of finding extreme solutions. Thus, they provide a wide range of alternative layout choices for the designers.

The required time for a complete evolutionary cycle mentioned in Table 2 also shows that the proposed method with adaptive VNS takes less time than the traditional local search-based approach, and the difference is very significant. For obvious reason, the proposed approach takes slightly more time than the approach without local search. Despite that it is very important to note that the performance of our proposed approach with adaptive VNS is much better than the other two approaches. In fact, the experimental results suggest that after performing half of the scheduled generations, the proposed approach with adaptive scheme starts finding the known best values for both objectives for more than 50% of the populations for all test problems. Where as, at this point the performances of the other two approaches are not satisfactory enough. They can find the best values only for small problems (up to 14-facility problems for traditional local search, and only up to 9-facility problems for without local search). The number of optimal solutions is also small. Table 2 summarizes the percentages of the optimal solutions obtained by each approach.

As mentioned earlier, we run the proposed approaches for 60% of the scheduled generations for all test problems to test their convergence behaviors. The experimental results suggest that at this stage, the proposed VNS-based approach almost convergences for around 70% of the total population. Fig. 8 also shows this tendency. For this reason, the required time for the proposed approach will be less than the time mentioned in Table 2. However, for fair comparison, we mention the time for the same number of generations for all approaches. Thus, the proposed adaptive VNS scheme appears to be highly effective, and the additional coding effort and time required in comparison to the approach without local search is definitely justified.

To summarize the result, the proposed evolutionary approach incorporating adaptive VNS for solving unequal area multi-objective FLPs is capable of producing near-optimal and non-dominated

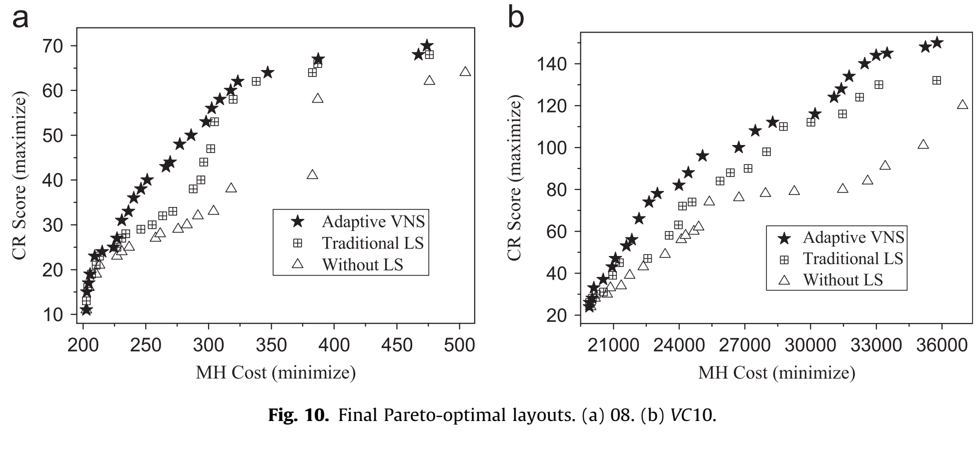

Fig. 10. Final Pareto-optimal layouts. (a) 08. (b) *VC*10.

layouts, which are also the best-known results in many cases. The results clearly show that the introduction of adaptive local search scheme helps well improving the best and average fitness value, average number of generations, convergence and diversity behavior of the final Pareto-optimal layouts.

In future, we hope to improve the proposed approach by considering several recently proposed faster and efficient MOEA variants. These could include multi-objective differential evolution based on the summation of normalized objectives and improved selection method (SNOV–IS) [43], Decomposition–based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes (ENS–MOEA/D) [44], and multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection (SNOV–DS) [45]. We would also like to analyze the performances of the final Pareto-optimal layouts using additional statistical tests, which are recently published in the literature [46].

7. Conclusion

The unequal area FLP has been an emerging topic in the recent years. A large volume of current research in unequal area FLPs has been conducted to satisfy the quantitative (distance-based) objective, while ignoring the aspect of adjacency (qualitative aspect) in the layout. In this paper, we developed an adaptive variable neighborhood search (VNS)-based evolutionary approach for solving the unequal area multi-objective FLP to find a set of Pareto-optimal layouts which is more relevant for practical use. The main reason for using the local search technique in multi-objective GA is to reinforce the search ability for locating globally optimal solution. However, this research goes one step further by incorporating a modified adaptive local search in form of the VNS using the modified SCM and the improved 1-*opt* local search method to automatically determine whether or not local search should be used in a GA loop. Consequently, this can save the computation time wasted by the traditional local search. The computational experiments strongly support the competitiveness of the proposed adaptive local search scheme for solving unequal area multi-objective FLPs in comparison to existing heuristic methods, approaches with traditional local search, and without local search. The proposed scheme is capable of finding a set of Pareto-optimal layouts that optimizes both MH cost and CR score simultaneously throughout the entire evolutionary process. Thus, it provides a wide range of alternative choices, allowing decision makers to be more flexible and to make better decisions based on market circumstances. The experimental results also shows that the proposed approach is more adept at improving the best and average fitness values, the required time, and the convergence behaviors of the trade-off solutions.

References

[1] N.N. Nordin, Z.M. Zainuddin, S. Salim, R.R. Ponnusamy, Mathematical modeling and hybrid heuristic for unequal size facility layout problem, Journal of Fundamental Sciences 5 (1) (2009) 87–89.
[2] S.S. Heragu, Facilities Design, PWS Publishing, Boston, MA, 1997.
[3] K.S.N. Ripon, K. Glette, M. Hovin, J. Torresen, A genetic algorithm to find pareto-optimal solutions for the dynamic facility layout problem with multiple objectives, in: K. Wong, B. Mendis, A. Bouzerdoum (Eds.), Neural Information Processing. Theory and Algorithms, Lecture Notes in Computer Science, vol. 6443, Springer, Berlin Heidelberg, 2010, pp. 642–651.
[4] A. Tompkins, Facilities Planning, 2nd edn, John Wiley& Sons, New York, 2003.
[5] J. Balakrishnan, C.H. Cheng, Multi-period planning and uncertainty issues in cellular manufacturing: a review and future directions, European Journal of Operational Research 177 (1) (2007) 281–309.
[6] J. Balakrishnan, C.H. Cheng, The dynamic plant layout problem: incorporating rolling horizons and forecast uncertainty, Omega International Journal of Management Science 37 (1) (2009) 165–177.
[7] P. Arikaran, V. Jayabalan, R. Senthilkumar, Analysis of unequal areas facility layout problems, International Journal of Engineering 4 (1) (2010) 44–51.
[8] S.P. Singh, R.R.K. Sharma, A review of different approaches to the facility layout problems, The International Journal of Advanced Manufacturing Technology 30 (5–6) (2006) 425–433.
[9] A. Drira, H. Pierreval, S. Hajri-Gabouj, Facility layout problems: a survey, Annual Reviews in Control 31 (2) (2007) 255–267.
[10] D.M. Tate, A.E. Smith, Unequal area facility layout using genetic search, IIE Transactions 27 (1994) 465–472.
[11] K.S.N. Ripon, K.N. Khan, K. Glette, M. Hovin, J. Torresen, Using pareto-optimality for solving multi-objective unequal area facility layout problem, in: N. Krasnogor, P.L. Lanzi (Eds.), Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation GECCO '11, ACM, New York, USA, 2011, pp. 681–688.
[12] T.A. Lacksonen, Static and dynamic layout problems with varying areas, The Journal of the Operational Research Society 45 (1) (1994) 59–69.
[13] N.R. Sanders, L.P. Ritzman, Improving short-term forecasts, Omega, International Journal of Management Science 18 (4) (1990) 365–373.
[14] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, 1st ed., Wiley, Chichester, 2001.
[15] P. Hansen, N. Mladenović, Variable neighborhood search: principles and applications, European Journal of Operational Research 130 (3) (2001) 449–467.
[16] N. Mladenović, P. Hansen, Variable neighborhood search, Computers & Operations Research 24 (11) (1997) 1097–1100.
[17] H. Ishibuchi, T. Yoshida, Implementation of local search in hybrid multi-objective genetic algorithms: a case study on flowshop scheduling, in: L. Wang, K.C. Tan, T. Furuhashi, J.H. Kim, X. Yao (Eds.), Proceedings of the 4th Asia–Pacific Conference on Simulated Evolution and Learning, SEAL 2002, Singapore, 2002, pp. 193–197.
[18] P. Merz, B. Freisleben, Greedy and local search heuristics for unconstrained binary quadratic programming, Journal of Heuristics 8 (2) (2002) 197–213.
[19] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[20] S.P. Singh, Solving facility layout problem: three-level tabu search metaheuristic approach, International Journal of Recent Trends in Engineering 1 (1) (2009) 73–77.
[21] G.C. Armour, E.S. Buffa, A heuristic algorithm and simulation approach to relative location of facilities, Management Science 9 (2) (1963) 294–309.
[22] A. Konak, S. Kulturel-Konak, B.A. Norman, A.E. Smith, A new mixed integer programming formulation for facility layout design using flexible bays, Operations Research Letters 35 (6) (2006) 660–672.
[23] I. Castillo, J. Westerlund, S. Emet, T. Westerlund, Optimization of block layout design problems with unequal areas: a comparison of MILP and MINLP optimization methods, Computers & Chemical Engineering 30 (1) (2005) 54–69.
[24] M.H. Hu, M.J. Wang, Using genetic algorithms on facilities layout problems, The International Journal of Advanced Manufacturing Technology 23 (3) (2004) 301–310.
[25] Q. Liu, R.D. Meller, A sequence-pair representation and MIP-model based heuristic for the facility layout problem with rectangular departments, IIE Transactions 39 (4) (2007) 377–394.
[26] D. Scholz, A. Petrick, W. Domschke, STaTS: a slicing tree and tabu search based heuristic for the unequal area facility layout problem, European Journal of Operational Research 197 (1) (2009) 166–178.
[27] Komarudin, K.Y. Wong, Applying ant system for solving unequal area facility layout problems, European Journal of Operational Research 202 (3) (2010) 730–746.
[28] S. Kulturel-Konaka, A. Konak, Unequal area flexible bay facility layout using ant colony optimisation, International Journal of Production Research 49 (7) (2011) 1877–1902.
[29] R. Logendrana, T. Kriausakula, A methodology for solving the unequal area facility layout problem using distance and shape-based measures, International Journal of Production Research 44 (7) (2006) 1243–1272.
[30] K.S.N. Ripon, K. Glette, O. Mirmotahari, M. Hovin, J. Torresen, Pareto optimal based evolutionary approach for solving multi-objective facility layout problem, in: C. Leung, M. Lee, J. Chan (Eds.), Neural Information Processing, Lecture Notes in Computer Science, vol. 5864, Springer, Berlin, Heidelberg, 2009, pp. 159–168.
[31] C. Grosan, A. Abraham, S. Tigan, T.-G. Chang, How to solve a multicriterion problem for which pareto dominance relationship cannot be applied? A case study from medicine, in: B. Gabrys, R. Howlett, L. Jain (Eds.), Knowledge-based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, vol. 4253, Springer, Berlin, Heidelberg, 2006, pp. 1128–1135.
[32] Y. Yun, C. Moon, D. Kim, Hybrid genetic algorithm with adaptive local search scheme for solving multistage-based supply chain problems, Computers & Industrial Engineering 56 (3) (2009) 821–838.
[33] P. Hansen, N. Mladenović, Variable Neighborhood Search, in: F. Glover, G. Kochenberger (Eds.), Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol. 57, Springer, New York, 2003, pp. 145–184.
[34] V. Roshanaei, B. Naderi, F. Jolai, M. Khalili, A variable neighborhood search for job shop scheduling with set-up times to minimize makespan, Future Generation Computer Systems 25 (6) (2009) 654–661.
[35] J.D. Witte, The use of similarity coefficients in production flow analysis, International Journal of Production Research 18 (4) (1980) 503–514.

[36] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm and Evolutionary Computation 1 (1) (2011) 32–49.
[37] C. Coello Coello, Recent trends in evolutionary multiobjective optimization, in: A. Abraham, L. Jain, R. Goldberg (Eds.), Evolutionary Multiobjective Optimization, Advanced Information and Knowledge ProcessingSpringer, Berlin, Heidelberg, 2005, pp. 7–32.
[38] A.E. Eiben, S.K. Smit, Parameter tuning for configuring and analyzing evolutionary algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 19–31.
[39] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, V. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.
[40] Y. Collette, P. Siarry, Three new metrics to measure the convergence of metaheuristics towards the Pareto frontier and the aesthetic of a set solutions in biobjective optimization, Computers and Operations Research 32 (4) (2005) 773–792.
[41] J. R. Schott, Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization, Master's Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.
[42] J. Wu, S. Azarm, Metrics for quality assessment of a multiobjective design optimization solution set, Journal of Mechanical Design 123 (1) (2001) 18–25.
[43] B.Y. Qu, P.N. Suganthan, Multi-objective differential evolution based on the summation of normalized objectives and improved selection method, in: 2011 IEEE Symposium on Differential Evolution (SDE), 2011, pp. 1–8.
[44] S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes, IEEE Transactions on Evolutionary Computation 16 (3) (2012) 442–446.
[45] B.Y. Qu, P.N. Suganthan, Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection, Information Sciences 180 (17) (2010) 3170–3181.
[46] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 3–18.