A NOVEL METASEARCH ALGORITHM FOR FACILITY LAYOUT OPTIMIZATION

Imran A. Tasadduq¹, M.H. Imam¹, Abdul-Rahim Ahmad²
¹Department of Computer/Civil Engineering, Umm Al-Qura University, Makkah, Saudi Arabia
iatasadduq@uqu.edu.sa, mmimam@uqu.edu.sa
²Systems Design Engineering, University of Waterloo, Waterloo, Canada
arahim@uwaterloo.ca

Abstract

Facility layout problem (FLP) is an inherently difficult combinatorial problem for which the quest for analytical, heuristic, and metaheuristic solution approaches has received significant attention. To this end, genetic algorithm (GA) has been an extensively studied metasearch approach. In this regard, we report a very effective memetic search algorithm (MA), which is a simple but novel variation of traditional GAs. The proposed MA is tested using a range of benchmark problems and results attest to its efficacy. Such effective search algorithms offer a significant promise in the quest of more efficient and effective decision aids for FLP. We conclude with some interesting future research directions.

Keywords

Layout design, facilities planning, genetic algorithms, memetic algorithms, metaheuristics

1 Introduction

Facility Layout Problem (FLP) is a well-studied combinatorial optimization problem arising in a variety of applications [5]. Often, the goal is to produce superior or optimal layouts for problems involving rectangular modules located so as to minimize a cost function without overlaps. The NP-complete nature of the problem means that, in almost all cases, the real optimal may not be known [1].

FLPs can be classified on various bases including: equal Vs unequal module areas, soft Vs hard modules, open Vs enclosed placement space. A review of different approaches reported in the literature for solving FLPs is given in [22]. FLP involving modules of equal area is usually referred to as cell assignment problem, where a certain area is divided into identical cells and each module is assigned to one of the cells. Several approaches have been used to solve unequal area FLP problems such as GAs, tabu search, neural networks, etc. [1, 11, 17]. In contrast, FLP with unequal area modules are considered more difficult and involve different solution approaches [13].

Unequal area module FLP can be classified on the basis of Soft or Hard nature of modules. A soft module has a pre-specified area but non-rigid dimensions [14]. In contrast, a hard module has rigid dimensions [25]. Although common in practice, these problems are harder to solve with relatively research efforts in this direction, making it an interesting domain.

FLPs may also be classified based on placement area, i.e. open Vs closed space. The enclosing area constraint may be imposed to obtain optimal layouts with a given rectangular, a single row or multiple rows [7, 21]. Whereas, FLP with open space imposes no constraint on the enclosing area [15]. Our research focuses on FLPs within open space involving hard and unequal area modules (OHU-FLP), which has received very little attention in past. Such FLPs have been solved using analytical, heuristic, and metaheuristic methods. Among analytical methods, various algorithms have been reported by [8, 10, 15, 16]. Notably, the novel idea of Cluster Boundary Tracking provides very remarkable results [8]. Metaheuristics like simulated annealing have been successfully applied to solve OHU-FLP [23]. Recently,
some evolutionary algorithms, mainly GAs have been proposed for OHU-FLP [6]. A good survey of GA
based algorithms for OHU-FLP along with benchmark problems is present in [13].

However, traditional GAs in OHU-FLP encounter such issues as unusable white space, multi-clusters, and
convergence to a single row. To overcome these issues, we present a simple yet very effective Memetic
Algorithm (MA) for solving OHU-FLP. MAs are a very promising technique in optimization but they
have not been used to solve OHU-FLP in past. MA is a form of hybrid global-local heuristic search
methodology where global search represents a GA while local search resembles a meme [20]. MAs greatly
improve the efficiency of search for the optimal solutions [4]. A good account of such MAs is [19].

This paper is organized as follows. Section 2 presents a mathematical formulation of OHU-FLP. Section 3
presents the proposed MA. Section 4 provides simulation results showing the efficacy of the proposed
algorithm. Section 5 concludes the paper with some interesting research directions.

2 Problem Formulation

Consider $N$ rectangular modules of fixed dimensions to be placed at their optimal positions in the
Euclidean plane without overlaps. For a module $i$, let the position be defined by its centroid $(x_i, y_i)$ and the
length and the width denoted by $L_i$ and $W_i$, respectively. Let $d_{ij}$ represent the cost of flow per unit distance
between centroids of modules $i$ and $j$. The objective is to minimize the cost function $C$, without any
overlap area $A_{ij}$ between two modules $i$ and $j$:

$$\text{Minimize } C(x_1, y_1, x_2, y_2, ..., x_N, y_N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} f_{ij}d_{ij}$$

Subject to

$A_{ij} \leq 0; \quad i = 1, 2, ..., N - 1; \quad j = i + 1$ to $N$

A positive $A_{ij}$ indicates there is an overlap else there is no overlap. It is defined as follows [15]:

$$A_{ij} = \lambda_{ij}(\Delta X_{ij})(\Delta Y_{ij})$$

where

$$\Delta X_{ij} = \left(\frac{L_i + L_j}{2}\right) - |x_i - x_j|$$

$$\Delta Y_{ij} = \left(\frac{W_i + W_j}{2}\right) - |y_i - y_j|$$

$$\lambda_{ij} = \begin{cases} -1 & \text{for } \Delta X_{ij} \leq 0 \text{ and } \Delta Y_{ij} \leq 0 \\ +1 & \text{otherwise} \end{cases}$$

3 Proposed Memetic Algorithm

While using GAs in OHU-FLP, it is important to get a user-specified accuracy for each decision variable
in each individual within a regular GA iteration [24]. This can be accomplished through addition and
subtraction operations on decision variables using user-specified increments. Similar operations are also
performed when GA converges to local minima with multiple disjointed clusters of modules. In such
situations, usual crossover or mutation operations do not improve the objective value. It requires all the
modules of cluster(s) move simultaneously so as to join other cluster(s), which is difficult to achieve in
conventional GAs.
The proposed algorithm has the added capability of detecting multiple clusters once the algorithm converges to a local minimum. Encountering multiplicity in clusters, the algorithm moves all the modules from the smallest cluster using operations similar to those described above, such that it joins another cluster. This process is repeated till we get a single cluster. Based on the elitist selection principle, only the fittest chromosome of the population is given the opportunity to perform addition or subtraction with the user-specified increment.

Interestingly, this incorporation of the user-specified accuracy performs an efficient local search prior to the next GA, rendering the algorithm its memetic categorization. Our literature search suggests that such an efficient local search is a novel contribution to the body of knowledge.

**Initial Population**

The algorithm starts with random initial population of \( P \) individuals/solutions using user specified accuracy and bounds. The randomly generated chromosomes represent the set of \( N \) genes consisting of decision variables such that the value of each decision variable specifies the lower-left corner of the corresponding module. In this initial population, overlaps are tolerated as these are automatically removed in the subsequent steps.

**Genetic Operations**

Next, the conventional crossover and mutation operations of GAs are performed. In crossover, the algorithm randomly chooses two individuals from the population. For a chromosome of length \( N \), it may choose a random position \( l \) such that \( l \leq N - 1 \). It then swaps all the genes at positions beyond \( l \) with a predefined probability of 0.8. For mutation, a mutation rate of 0.02 is used. A randomly selected gene is replaced with a randomly chosen location point within upper and lower bounds. The selection of the new location point is based on uniform distribution.

**Memetic Operation**

Next, a local search using addition and subtraction operations as per user-specified increments takes place on the fittest chromosome in the GA generation. First, an addition operation is performed on the chromosome, resulting in a child. Next, a subtraction operation is performed on the chromosome, resulting in another child. These operations are performed on all decision variables resulting in \( 2N \) offspring. If the fittest of these offspring is superior to the fittest individual in the GA generation then it is retained in the population for the next iteration of MA. This memetic operation results in an efficient local search within the global search done by GA.

Since, the fittest chromosome of a generation performs addition and subtraction with user-specified increment, \( 2N \) offspring are born. Thus, we will now have \( 2N+1 \) chromosomes in the pool, as the fittest is added to the original population and others are discarded.

**Selection Operation**

The fitness values are used to select a subset of the current population based on the elitist principle of “survival of the fittest”, where individuals having a better fitness value survive while the rest perish. A number of selection strategies have been proposed in the literature, such as, fitness-proportional selection, ranked selection, tournament selection, truncation selection, and mutation and recombination. We used widely accepted roulette wheel selection mechanism.

**Termination Criteria**

We terminate our algorithm when the improvement in the objective function between two consecutive generations is less than \( 10^{-6} \).
The pseudo code of the complete algorithm is shown in Table-1 as well as depicted in Figure-1.

<table>
<thead>
<tr>
<th>Proposed Memetic Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Specify upper and lower bounds, and desired increment</td>
</tr>
<tr>
<td>2 Initialize population</td>
</tr>
<tr>
<td>3 Evaluate population</td>
</tr>
<tr>
<td>4 While termination criterion not reached</td>
</tr>
<tr>
<td>5 Perform crossover and mutation</td>
</tr>
<tr>
<td>6 Perform addition and subtraction between fittest chromosome and user-specified increment and generate two children</td>
</tr>
<tr>
<td>7 Evaluate population</td>
</tr>
<tr>
<td>8 Select solutions for next population</td>
</tr>
<tr>
<td>9 Detect if there are multiple clusters</td>
</tr>
<tr>
<td>10 While there are multiple clusters</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

| Table 1: Proposed MA (Pseudo Code) | Figure 1: Proposed MA (Flow Chart) |

4 Simulation Results

We have used published benchmark problems for comparison of the performance of the proposed MA with the best available reported results that can be verified. In addition, a commercially available layout optimization software named VIP-PLANOPT™, which is known for quickly providing remarkably superior solutions, is used to generate solutions to benchmark problems for our comparative analyses. Data specific to the test problems and best available verifiable solutions are listed in [24]. It may be noted here that some authors have claimed good solutions to several published problems without providing any data to verify the claimed best solutions. Since there is no way to verify such layouts, such published claims have not been included in our comparisons.

In Table-2, we provide solutions for various benchmark problems with column-2 and column-3 indicating the size and the known original source of the problem, respectively. Columns 4-6 provide verifiably best solutions from published sources, VIP-PLANOPT, and the Proposed MA, respectively. Despite some published claims, most problems do not have an actual verifiable solution in published sources. Indeed, various claimed best solutions can be proven incorrect or impossible [24].

It can be easily seen that most published problems do not have any reported solutions that can be verified through the use of data regarding module locations. Furthermore, VIP-PLANOPT (www.planopt.com) provides some very good solutions using a propriety algorithm. The last column in Table-2 contains the solutions obtained through the proposed MA. It can be seen that the proposed MA matches or surpasses
the available best solution in most instances, indicated by an asterisk. It shows that the proposed MA is significant contribution to the body of knowledge both from practical and commercial perspectives.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Size (Modules)</th>
<th>Original Source</th>
<th>Verifiably Best Available Solutions Published</th>
<th>VIP-PLANOPT</th>
<th>Proposed MA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>[11]</td>
<td>270</td>
<td></td>
<td>270*</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>[3]</td>
<td>-</td>
<td>1,510</td>
<td>1,510*</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>[3]</td>
<td>-</td>
<td>3,379</td>
<td>3,314.8*</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>[8]</td>
<td>792.43 [17]</td>
<td>692.5</td>
<td>689.5*</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>[25]</td>
<td>-</td>
<td>763.5</td>
<td>757.5*</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>[3]</td>
<td>-</td>
<td>18,752</td>
<td>19,279</td>
</tr>
<tr>
<td>9</td>
<td>50</td>
<td>[25]</td>
<td>80,794.24 [15]</td>
<td>78,224.7</td>
<td>77,504*</td>
</tr>
<tr>
<td>10</td>
<td>62</td>
<td>[6]</td>
<td>-</td>
<td>3,996,206</td>
<td>4,278,682</td>
</tr>
<tr>
<td>12</td>
<td>125</td>
<td>[25]</td>
<td>-</td>
<td>1,084,451</td>
<td>1,099,290</td>
</tr>
</tbody>
</table>

Table 2: Summary of simulation results

5  Conclusions

We propose a novel, efficient, and effective memetic algorithm (MA) for solving unconstrained, unequal area, hard module facility layout problems. The proposed MA is based upon representing the decision variables in terms of user-specified accuracy, which reduces the overall computational cost. The proposed implementation brings substantial, often drastic, improvement by addressing the low probability of small variations in the decision variable in a traditional GA. It also overcomes the Hamming Cliff problem by imposing the smallest variation in each decision variable in each generation. Simulation results clearly demonstrate the efficacy of the proposed algorithm. Such effective search algorithms offer a significant promise in the quest of more efficient and effective decision aids for automated facility layout design optimization. In future, we want to research the effects of using an adaptive local search increment in the proposed algorithm, as opposed to constant increment used in the reported MA. Use of hybrid search mechanism in conjunction with some effective placement algorithms will also be explored.

Bibliography