Fixed and flexible shape facility layout problems using biogeography-based optimisation algorithm

Supriya Ingole and Dinesh Singh*

Department of Mechanical Engineering,
Sardar Vallabhbhai National Institute of Technology,
Surat, Gujarat, India
Email: supriya.ingole@rediffmail.com
Email: dineshsinghmed@gmail.com
*Corresponding author

Abstract: For any manufacturing and production industry to be efficient in terms of revenue, the layout of the industry must be arranged to minimise the total material handling and transporting cost of that industry. All the facilities, machines, storage, etc. are allocated in the layout in an optimum manner. While designing a layout, the planner has to allocate equal or unequal shape facilities on the available layout space to minimise the distance between the facilities which are having maximum material flow. The problem concerned with assigning facilities to the floor is mentioned as facility layout problem (FLP). This paper focuses on unequal area FLPs with fixed shape and flexible shape facilities. Biogeography-based optimisation (BBO) algorithm is one of the metaheuristic methods emerged in recent times to cope with various optimisation problems. Therefore, BBO is considered to solve unequal area FLPs. The results obtained using BBO algorithm are compared and found better than previous known results.

Keywords: unequal area facility layout; fixed-shape facilities; flexible-shape facilities; biogeography-based optimisation algorithm.


Biographical notes: Supriya Ingole received her BE in Mechanical Engineering in 2005 at Nagpur University, Maharashtra, India and MTech degree in CAD/CAM from Nagpur University, Maharashtra, India in 2012. She is a PhD research scholar in the Department of Mechanical Engineering of Sardar Vallabhbhai National Institute of Technology, Surat, India. She has two research paper published in international journals. Her area of research is facility layout design and non-traditional optimisation techniques.

Dinesh Singh received his BTech in Industrial Engineering from IIT Roorkee, India in 2001, MTech degree in Production and Industrial Systems Engineering from IIT Roorkee, India in 2006 and PhD in Mechanical Engineering from Sardar Vallabhbhai National Institute of Technology, Surat, India in 2012. He has more than 11 years of teaching and research experience and is currently working as an Assistant Professor in the Department of Mechanical
1 Introduction

In today’s changing environment of manufacturing and production fields, the facility layout design is a crucial stage and should be given great importance because an efficient layout can minimise the overall manufacturing cost. Facility layout problem (FLP) is concerned with the allocation of facilities or machines on the available floor space to optimise total material handling cost (MHC)/total layout cost. Anjos and Vieira (2017) briefly reviewed three types of layouts problems i.e., row layouts, unequal area layouts and multi-floor layouts. The authors presented the tight bounds in mixed integer programming (MIP) formulation to solve the layout problems. They explained symmetry breaking constraints and valid inequalities while solving FLPs.

There are various methods of representing the layouts. Such as equal area FLPs can be represented by blocks of same size and shape on the layout floor and can be formulated as quadratic assignment problem (QAP). Tosun et al. (2013) implemented GA with a number of crossover operators for solving different FLPs from QAPLIB. Zhou et al. (2016) developed a backtracking search algorithm to solve the FLPs from QAPLIB. Unequal area FLPs can be represented in different ways, such as, fixed shape structure, flexible bay structure (FBS), slicing tree structure, etc. Flexible shapes FLPs modelled using linear programming (LP) are solved by Kulturel-Konak (2012) with a probabilistic tabu search (TS) technique. These FLPs can be formulated as QAP or MIP problem. In fixed shape structure, the length, width and area of the facilities are fixed whereas in FBS the length and width of the facilities are flexible and the area is fixed. The length and width is given as input to the problem in fixed shape FLPs. In FBS, The width of the rows/bays of the layout is flexible therefore the inputs given to the problem are areas of facilities and their aspect ratio. The facilities are placed on the flexible bays with the constraint of maximum allowable aspect ratio. A penalty function is added to the objective function if the constraint is violated. The width of the facilities in a bay is equal to the total width of that bay. The main goal in solving FLP is to minimise the workflow cost and distance between the facilities. Mostly, the FLPs are formulated as QAP.

In this paper, biogeography-based optimisation (BBO) algorithm is used to solve unequal area FLPs. As a metaheuristic, BBO is a population-based algorithm which imitates the behaviour of species on different islands. This algorithm works on island theory which shares good features of an island with other islands and thereby improves the solution. Simon (2008) introduced and applied BBO algorithm on benchmark functions and showed that its performance is at par with other population-based methods such as genetic algorithm (GA), particle swarm optimisation (PSO), ant colony
optimisation (ACO), etc. Further, Rahmati and Zandieh (2012) applied BBO algorithm to solve flexible job shop scheduling problems and provided promising solutions. Singh and Ingole (2019) considered both quantitative and qualitative factors to solve equal area FLPs. They implemented the weight and Pareto approach for solving multi-objective FLPs and obtained the optimum results using BBO algorithm. In this paper, BBO algorithm is implemented to solve the unequal area FLPs due to its merits and applicability to solve such types of problems. In the next section, the literature review on unequal area FLPs is presented.

2 Literature review

Over the last few decades, the FLPs have received extensive consideration. Researchers focused nearly on all the categories of FLPs. It includes equal and unequal area FLPs, static and dynamic FLPs, single and multi-objective FLPs, etc. A comprehensive literature review on facility layout design is given by Meller and Gau (1996), Singh and Sharma (2006), and Drira et al. (2007). Recently, Hosseini-Nasab et al. (2018) presented various research trends in FLPs which include most of the aspects of FLPs. In this section, the attention is given on background and related work on unequal area FLPs along with the heuristic and metaheuristic techniques used to solve them effectively.

In general, equal area facility layouts cannot be considered as potential in all the cases of layouts due to different shape constraints on facilities. Therefore, a layout planner should opt for unequal area facility layouts to accommodate the facilities of different shapes and sizes. The presented literature review is categorised as unequal area fixed-shape FLPs and unequal area flexible-shape FLPs.

2.1 Unequal area fixed-shape FLPs

The articles in literature which approached the concept of unequal area fixed-shape FLPs are depicted here. Schnecke and Vornberger (1997) implemented a hybrid genetic algorithm (HGA) to solve unequal area fixed-shape FLPs with a tree structure. Lee and Lee (2002) presented unequal area fixed shape FLPs and employed HGA to find the optimal solutions. Scholz et al. (2009) presented a TS heuristic to solve fixed and flexible shape unequal area FLPs using slicing tree representation. Hakobyan and McKendall (2013) presented a LP model to attempt unequal area fixed-shape dynamic FLPs and optimised using hybrid method which consists of dual simplex method and TS algorithm. Ingole and Singh (2017) attempted firefly algorithm (FA) to solve benchmark FLPs and provided the solutions for randomly generated large size FLPs to check the effectiveness of FA. Park and Seo (2017) presented a two-step heuristic algorithm consists of construction and improvement steps to solve fixed dimension unequal area FLPs. They placed the facilities in horizontal and vertical groups and generated the input/output points. Feng and Che (2018) attempted fixed shape rectangular FLPs and reformulated using integer LP with new constraints to maximise the material flow between the closed facilities.
2.2 Unequal area flexible-shape FLPs

One of the most studied methods to represent the layout sequence or arrangement is FBS which was introduced by Tate and Smith (1995). In FBS, the rectangular floor space is divided into bays or rows of different widths in horizontal or vertical direction, and each bay is divided into one or more departments having same width and varying length. Tate and Smith (1995) implemented GA to unequal area FLPs with FBS. Konak et al. (2006) used FBS representation for solving unequal area FLPs with MIP formulation. In MIP, a constraint is imposed on widths of the bays based on the departments assigned to each bay and another constraint restricts the height of the departments between the permissible side lengths. A sequence-pair representation is introduced by Liu and Meller (2007) to solve unequal area FLPs using MIP formulation and optimised using GA.

Kulturel-Konak and Konak (2011a) proposed ACO approach for solving unequal area FLPs with FBS. Kulturel-Konak and Konak (2011a) introduced a relaxed FBS to solve FLPs using PSO. Jankovits et al. (2011) presented a convex-optimisation framework to find optimal solutions of unequal area FLPs, where the framework combines two mathematical programming models. Kulturel-Konak (2012) proposed a probabilistic tabu search (PTS) technique to solve LP-based FLPs. An encoding of layout representation for unequal area FLPs with FBS was introduced by Ulutas and Kulturel-Konak (2012) and optimised the large size problems using clonal selection algorithm (CSA). Mazinani et al. (2013) attempted dynamic FLPs up to 12 facilities based on FBS and optimised using GA. Abdezadeh et al. (2013) presented a MIP formulation for multi-objective dynamic FLP concerning FBS and parallel variable neighbourhood search algorithm was used to obtain optimum results. HGA with LP approach to solve FLPs was introduced by Kulturel-Konak and Konak (2013). Garcia-Hernandez et al. (2013) optimised the layouts of three industrial case-studies using an Interactive GA considering FBS FLPs which consist of some subjective features from the knowledge of designer. Anjos and Vieira (2016) introduced a framework for solving the unequal area FLPs by combining two mathematical optimisation models. Palomo-Romero et al. (2017) proposed an island model genetic algorithm (IMGA) to attempt FLPs based on FBS. Asef-Vaziri et al. (2017) reported the results of unequal area FLPs having FBS based on loop layout and developed the model for the same. The model consists of material flow with loaded and empty trips between the input/output stations in unidirectional circular flow paths. The authors first find the promising layouts and then improved the algorithm to find actual function values to save the computational time. Moslemipour et al. (2018) proposed a hybrid algorithm combining ant colony, clonal selection and SA to solve large size FLPs of stochastic and deterministic in nature. Abbasi et al. (2017) designed facility layout on the basis of mathematical model considering relation between the facilities and attempted a case study of hospital by considering three quantitative criteria (material handling, neighbouring connection and shape ratio of facilities) and solved using CPLEX 12. Agarwal and Singholi (2018) considered the layout of axle assembly division of a company to reach at alternative optimal solutions using AHP and fuzzy AHP to evaluate the weights of the criteria and TOPSIS or fuzzy TOPSIS to rank the alternative layouts.
Apart from FBS, other methods to represent layouts are also addressed by some of the researchers. One of the representations of unequal area facility layout is slicing tree structure. In slicing tree representation, the facility floor is repeatedly divided in vertical and horizontal directions to arrange the facilities. The node of the slicing tree indicates the direction in which the facilities to be placed (Tam, 1992). Tam (1992) described the unequal area FLP with slicing tree structure and considered shape constraints such as aspect ratio and dead space ratio of each department and obtained the solutions using simulated annealing (SA) algorithm. Kim and Kim (1995) considered unequal area FLPs and proposed a graph theoretic heuristic, in which the initial layout were generated using maximal planar adjacency graph and obtained the improved layout through local search technique. Gau and Meller (1999) formulated FLPs as MIP and implemented GA with three different approaches, i.e., single tree structure, multiple-tree structure and addition of dummy departments to the layout. Krishnan et al. (2006) used dynamic material flow between departments which track the flow in continuous manner to save the cost of the rearrangement of the layout and analyse the flow over time periods for dynamic FLPs. Wilsten and Shayan (2007) considered a FLP of furniture manufacturing company and optimised it using various approaches like GA, computerised relative allocation of facilities technique (CRAFT), construction algorithm, automated algorithm, etc. and compared the obtained results using analytical hierarchy process (AHP). Komarudin and Wong (2010) solved unequal area FLPs with slicing tree representation by ant system (AS). Gress et al. (2011) analysed the use of GA with local search to optimise unequal area FLP using binary variables in MIP formulation. Bozer and Wang (2012) introduced a graph-pair representation to solve unequal area FLP formulated as MIP considering overlapping constraint and optimised it using SA algorithm. Goncalves and Resende (2015) developed GA based on biased random number with new encoding for unequal area FLPs. Xiao et al. (2017) developed a problem evolution algorithm (PEA) combined with LP to obtain solutions for unequal area FLPs and zone-based dynamic FLPs. Kang and Chae (2017) presented the harmony search algorithm to obtain best solutions for unequal area FLPs. They represented the layout as slicing tree structure and modified it and readjusted to diversify the potential range of solutions.

From above discussion, it is observed that more attention has been given on metaheuristic/optimisation techniques like GA, PSO, ACO, etc. to find the near optimal solutions of FLPs; as these techniques provide better results in reasonable time. The BBO algorithm has some common features similar to GA and PSO and it have flexibility to attempt various types of problems, for e.g., Lim et al. (2016) proposed a hybrid technique combining BBO and TS to solve QAP and obtained the better results within reasonable computational time. Sooncharoen et al. (2015) described the application of BBO for machine layout design problems to minimise the material handling distance. Therefore, in this article, BBO algorithm is implemented, to solve unequal area FLPs.

From the literature survey, it is observed that BBO algorithm is not attempted to solve unequal area FLPs. Further, unequal area flexible-shape FLPs having large number of departments are attempted by many researchers, however fixed-shape FLPs of large size are not attempted. Therefore the main objective of this study is to implement BBO algorithm to solve unequal area FLPs having fixed-shape and flexible-shape facilities.
3 Research methodology

Non-traditional optimisation or metaheuristic techniques are approximate solution techniques. These techniques have been used since the beginning of operations research to tackle difficult optimisation problems of continuous and discrete functions. These optimisation algorithms are effectively applicable in almost all the fields of engineering optimisation. Non-traditional optimisation techniques require less computational time as compared to traditional methods and have great potential to provide near optimal solutions. There are numerous non-traditional optimisation techniques which are invented and implemented successfully in various applications including FLPs.

FLP belongs to the non-polynomial hard (NP-hard) class problem believed to be unsolvable in polynomial time. Heuristics or metaheuristics are usually implemented to solve these NP-hard problems. The exact solution can be obtained from optimal methods in a reasonable time only when the problem size is small. It has been revealed that the computational times for solving FLPs are likely to increase exponentially as the number of facilities to be located gets increased. Therefore, the non-traditional optimisation techniques are most suitable to obtain the optimal solutions for large size FLPs. It is observed from literature review that BBO algorithm is capable to handle complex optimisation problems. The migration and mutation operators of BBO algorithm works very well for the near optimal convergence of the algorithm which results into better performance than other algorithms. Therefore, in present research work, BBO algorithm is attempted to obtain optimal solutions of FLPs.

3.1 Introduction to BBO

BBO is a population-based evolutionary algorithm and it is developed by Simon (2008). BBO is based on theory of island biogeography in which geographical distribution of biological species is explained (Alroomi et al., 2013). BBO algorithm follows migration and mutation operations to reach at global minimum solution. The mathematical model of BBO algorithm is developed by considering the migration behaviour of species from one island to another. On an island or habitat, if the living conditions are appropriate for species then that habitat have high habitat suitability index (HSI), because this habitat have better features than other habitats. The variables that characterise habitability of an island are called as suitability index variables (SIVs) (Simon, 2008). Habitats having high HSI have more number of species than that of low HSI habitats. So, the species on the high HSI habitats can immigrate to other habitat and have low species immigration rate as they are already full with species. Similarly, the low HSI habitats have high immigration rate of species because of their low population.

3.2 BBO algorithm

BBO algorithm is based on the concept of distribution of species on islands which converts into a general problem function solution. Each island is considered as a member or a solution. By considering the emigration rate (\(\mu\)) and immigration rate (\(\lambda\)) of each
member, the information between the habitats is shared probabilistically. High-\textit{HSI} habitat represents a good solution and low-\textit{HSI} habitat represents poor solution. The good solution means the island has lots of good features such as trees, food, rainfall, temperature, humidity, etc. Hence this island or habitat has high \textit{HSI}. Each feature is indicated as \textit{SIV}, which denotes the independent variable of the problem function or facilities in case of FLPs. Good solution features emigrate from high-\textit{HSI} islands to low-\textit{HSI} islands (Rahmati and Zandieh, 2012). Poor solutions accept several features and information from good solution, which further improves the solutions in the population of islands. In BBO, there is an elitism strategy, which preserves best members from the population in each iteration or generation. If the algorithm traps in local minima, then the elitist solutions will remain intact and gives the near-optimal solutions.

\subsection*{3.2.1 Algorithm Steps of BBO}

\textbf{Step 1} Initialisation of BBO parameters

\begin{itemize}
  \item \textbf{S} number of islands i.e. number of layouts; each island represents the permutation of facilities from \( n = 1, 2, \ldots, k \)
  \item \textbf{G} generations of algorithm
  \item \textbf{I} maximum immigration rate (Figure 1)
  \item \textbf{E} maximum emigration rate (Figure 1)
  \item \textbf{S}_{\text{max}} maximum number of species on an island.
\end{itemize}

\textbf{Step 2} Selection of the habitats for migration

In migration operator, good features or information is shared between habitats or islands. This depends on emigration rate \( \mu \) and immigration rate \( \lambda \) of each solution. Migration is performed by interchanging of facilities between the population members or habitats. The objective function value (OFV) or \textit{HSI} of each island is calculated. In FLPs, the OFV is minimised; hence low-\textit{HSI} is a good solution. In Figure 1, immigration curve is indicated as \( I \), which follows when there are no species on the island. The maximum number of species that an island can keep is \( S_{\text{max}} \), at which point the immigration becomes zero. Now, considering the emigration curve indicated as \( E \). If no species present on an island then the emigration will be zero. When an island holds the largest number of species, then the maximum emigration is \( E \). The emigration rate \( \mu_s \) and immigration rate \( \lambda_s \) in the presence of \( s \) species on that island are calculated from the equations (1) to (2) and migration starts.

\begin{align*}
\mu_s &= \frac{E}{S_{\text{max}}} S \\
\lambda_s &= 1 \left( 1 - \frac{s}{S_{\text{max}}} \right)
\end{align*}
Now the probability of existence of $S$ species on the island is calculated, which is denoted by $P_S$. This probability is obtained from equation (3), as the number of species varies from time $t$ to $t + \Delta t$.

$$P_S(t + \Delta t) = P_S(t)\left(1 - \lambda_S \Delta t - \mu_S \Delta t\right) + P_{S+1} \lambda_{S+1} \Delta t + P_{S-1} \mu_{S-1} \Delta t$$  \hspace{1cm} (3)

From equation (3), one of the following three conditions should satisfy to contain $S$ species on an island at time $(t + \Delta t)$:

1. $S$ species at time $t$ and no immigration or emigration took place during the interval $\Delta t$
2. $(S - 1)$ species at time $t$, and one species immigrated
3. $(S + 1)$ species at time $t$, and one species emigrated.

For finding $P_s(t)$ in a steady state, equation (4) is given:

$$P_s = \sum_{i=1}^{\max S + 1} v_i$$  \hspace{1cm} (4)

where $v$ and $v_i$ can be calculated from equations (5) and (6).

$$v = \left[v_1, v_2, \ldots, v_{\max + 1}\right]^T$$  \hspace{1cm} (5)

$$v_i = \frac{S_{\max}!}{(S_{\max} + 1 - i)(i - 1)!} \quad \left(i = 1, \ldots, S_{\max} + 1\right)$$  \hspace{1cm} (6)
Algorithm 1  Migration loop of BBO algorithm

Select an island $H_i$ in which the species will immigrate using probability $\lambda_i$ ($i = 1, 2, \ldots, s$)

If $\text{rand} < \lambda_i$
    for $j = 1$ to $k$
        Select an island $H_j$ from which the species will emigrate using probability $\mu_j$
        If $H_j$ is selected
            Randomly select an SIV or department from $H_j$
            Replace a random SIV or department in $H_i$ with that of $H_j$
        end if
    end for
end if

go to next SIV or department

go to next Island

After selecting immigrating and emigrating islands from initial population, the migration is done using the probabilities. From emigrating islands some of the facilities are migrated to immigrating islands and modified islands are obtained.

Step 3  Selection of the habitats for mutation.

The mutation operator is used to increase the diversity of the population members to obtain the good solutions. In island principle, the number of species present at equilibrium state can be differed due to some external events such as diseases, tsunamis, volcanoes or earthquakes which cause decrease in total number of species. If there are other suitable events which provide good features to an island, the solution gets improved (Alroomi et al., 2013). In BBO algorithm, mutation is done based on probability of species $P_s$ and is used for modifying one or more randomly selected SIV or facilities. The mutation rate can be obtained from equation (7).

$$m = m_{\max} \left(1 - \frac{P_s}{P_{\max}} \right)$$ (7)

where $m_{\max}$ is a user-defined maximum mutation rate that $m$ can reach, and $P_{\max}$ is the maximum species probability.

Algorithm 2  Mutation loop of BBO algorithm

For $i = 1:k$ (k is the number of islands)
    Calculate mutation rate ($m$) using equation (7)
    Select an island $H_i$ with probability $P_s$ for mutating SIV
    If $H_i$ is selected for mutation
        Replace selected SIV of island with a randomly generated SIV
    end if
end for
Step 4 Calculate the OFV of each population member and find best solution (minimum value) and corresponding layout.

Step 5 Repeat steps 2 to 4 until optimum value is obtained or maximum number of iterations are reached.

The flowchart of BBO algorithm is given in Figure 2.

**Figure 2** Flowchart of BBO algorithm
4 Mathematical formulation of unequal area FLP

4.1 Unequal area fixed-shape FLP

In most of the manufacturing industries, the shapes of departments/facilities are unequal; therefore the concept of unequal area fixed shape FLPs is approached by some researchers (Schnecke and Vornberger, 1997; Lee and Lee, 2002; Scholz et al., 2009). To minimise the overall transportation cost in a manufacturing layout, the objective function is depicted as given in equation (8). Lee and Lee (2002) have given the procedure of arrangement of facilities in layout for fixed-shape FLPs. The total length \( L \) and width \( W \) of the layout is known.

\[
\text{Min } F = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} f_{ij} d_{ij}
\]  

subject to

\[
\sum_{u=1}^{k} l_u \leq 1.25 \sqrt{\sum_{i=1}^{n} A_v}
\]  

where

- \( F \) is the total material handling/transportation cost
- \( i, j \) are facility indices (1, 2, ..., \( n \))
- \( n \) is the number of facilities/departments
- \( c_{ij} \) is the cost of one unit of material flow between facilities \( i \) and \( j \)
- \( f_{ij} \) is the amount of material flow between facilities \( i \) and \( j \)
- \( d_{ij} \) is the rectilinear distance between locations of facilities \( i \) and \( j \)
- \( l_u \) is the length of the facility
- \( A_v \) is the area of the facility
- \( u \) is a facility index for number of departments in a bay (1, 2, ..., \( k \))
- \( v \) is a facility index for number of departments (1, 2, ..., \( n \)).

The constraint given in equation (9) is used to assign the facilities on the layout in horizontal direction. The facilities are placed in the first bay one by one until the constraint (9) is satisfied else next facility is placed in second bay and the procedure is repeated till all the facilities get placed. The width of each bay is the maximum width of the facility in that particular bay (Lee and Lee, 2002; Ingole and Singh, 2017).

4.1.1 Layout representation for fixed shape FLP

The facilities can be arranged in horizontal or vertical direction on the floor space. Here, they are assigned in horizontal direction from top left corner to bottom right corner of the layout. The procedure of assignment of facilities on the floor space is described by Lee and Lee (2002). For e.g., the layout string 7-9-1-3-5-6-4-8-2 will be assigned as presented in Figure 3. The layout arrangement steps are as follows:
Step 1  Assign the facilities from top left in horizontal direction. The total width ($W$) and total length ($L$) are considered as arbitrary (Lee and Lee, 2002). The facilities are placed in the considered bay until the sum of lengths of facilities in the same bay is satisfied the constraint given in equation (9). If sum of lengths of facilities in a bay exceeds $L$, then new bay is created and facilities are placed in the new bay (Figure 3).

Step 2  Determine the width of the bay based on already positioned facilities. Width of first bay is equal to the width of the facility which has largest width in that bay.

Step 3  Repeat steps 1 and 2 until all the facilities are assigned as shown in Figure 3.

**Figure 3**  Representation of fixed shape facility layout (see online version for colours)

4.2 Unequal area flexible-shape FLP

The objective function to solve unequal area FLPS is to minimise total $MHC$ consists of material flow and the distance between the facilities. The bays/rows are of varying width in flexible shape facility layout representation. The facilities of known areas are allocated
in flexible bays. An example of FBS layout is depicted in Figure 4. The facilities are arranged from top to bottom starting from left side of the layout. In the first bay, facilities 3 and 6 are placed; in the second bay facilities 5, 10 and 7 are placed; in the third bay, facilities 9, 4 and 1 are placed and finally in the fourth/last bay, facilities 8 and 2 are placed. The width of each facility is calculated after assigning the facilities in the first bay. The width is the ratio of the total area of the facilities in the respective bay and the height of the layout as stated in equation (10). The width of the bay is equal to the width of the facilities in that bay and the length of the facility is determined by dividing the area of that facility to its width. After assigning the facilities in the first bay, the width of each facility in that bay can be calculated by dividing the total area of the facilities in that bay to the height of the total layout.

$$\text{Width of the bay} = \frac{\text{Total area of facilities in the bay}}{\text{Height of the layout}}$$

(10)

Figure 4 An example of FBS layout

The constraints of maximum allowable aspect ratio and minimum side length imposed on FLPS with FBS are presented by Ulutas and Kulturel-Konak (2012). A penalty cost function is considered to penalise the infeasible facilities that does not satisfy the constraint of maximum aspect ratio or minimum side length. The objective function is given as follows.

$$F(s) = MHC(s) + P_c(s)$$

(11)

$$MHC(s) = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{ij}$$

(12)

Subject to

$$\alpha_i = \frac{\max\{l_i, w_i\}}{\min\{l_i, w_i\}} \leq \alpha_{\text{max}}$$

(13)

$$l_i^{\text{min}} \leq \min\{l_i, w_i\}$$

(14)

$$P_c(s) = (n_c)^{\frac{1}{w}} \times MHC(s)$$

(15)

where
**5 Application of BBO algorithm to unequal area FLPs**

The proposed BBO algorithm is programmed in MATLAB 2010 and run on Microsoft Windows 8, Intel core processor with 4GB RAM. The demonstration steps of BBO algorithm with an illustrative example of 5 departments are stated in Appendix. In this section, five problems of fixed-shape facilities of size 15, 20, 30, 40 and 50 are attempted using BBO algorithm. Further, 11 problems of flexible-shape facilities, viz. O7, O8, O9, VC10Es, VC10Ea, VC10Rs, VC10Ra, BA12, BA14TS, Tam20 and Tam30 are attempted using BBO algorithm.

5.1 Parameter tuning of BBO algorithm

Proper tuning of parameters is very important to get the best results using a metaheuristic technique. For setting the parameter values, several trials of algorithm required to be carried out on FLPs. The setting of the parameter values depends on the nature of the problem. It may be stated that a specific range of parameter values of algorithm provides better solutions which can be used for further experiments. Several trials are taken to set the parameters of BBO algorithm for all the FLPs considered in this paper. The setting of BBO parameters is presented in Table 1. It is seen that the value of parameter ‘S’ which is the population of islands is minimum for small size FLPs as compared to large size FLPs. The generations required to solve large size FLPs are also high. Since, the solution space of large size problems is tremendously high; the algorithm requires more number of population and generations to reach at optimal solution. Other parameters such as maximum emigration and immigration can be considered in between 0.1 to 1. The range of mutation rate can be set between [0.1, 1], however it is kept below 0.4 because it may
create a random search in the space which may be difficult for better convergence of the algorithm to near optimal or best solution.

**Table 1**  
**BBO parameters setting**

<table>
<thead>
<tr>
<th>Type of FLP</th>
<th>Problem</th>
<th>No. of islands (S)</th>
<th>Generations (G)</th>
<th>Emigration (E)</th>
<th>Immigration (I)</th>
<th>Mutation rate (mmax)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed-shape FLPs</td>
<td>n = 15</td>
<td>40</td>
<td>100–200</td>
<td>0.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>n = 20</td>
<td>40</td>
<td>100–200</td>
<td>0.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>n = 30</td>
<td>40</td>
<td>100–200</td>
<td>0.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>n = 40</td>
<td>40</td>
<td>100–500</td>
<td>1.0</td>
<td>1.0</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>n = 50</td>
<td>40</td>
<td>100–500</td>
<td>1.0</td>
<td>1.0</td>
<td>0.3</td>
</tr>
<tr>
<td>Flexible-shape FLPs</td>
<td>O7</td>
<td>100</td>
<td>200</td>
<td>0.9</td>
<td>0.9</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>O8</td>
<td>100</td>
<td>200</td>
<td>0.9</td>
<td>0.9</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>O9</td>
<td>200</td>
<td>500</td>
<td>0.9</td>
<td>0.9</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>VC10Es</td>
<td>100</td>
<td>500–1,000</td>
<td>0.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>VC10Fa</td>
<td>100</td>
<td>500–1,000</td>
<td>0.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>VC10Rs</td>
<td>100</td>
<td>500–1,000</td>
<td>0.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>VC10Ra</td>
<td>100</td>
<td>500–1,000</td>
<td>0.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>BA12</td>
<td>100</td>
<td>5,000</td>
<td>0.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>BA14TS</td>
<td>150</td>
<td>5,000</td>
<td>1.0</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>Tam20</td>
<td>250–300</td>
<td>5,000–8,000</td>
<td>1.0</td>
<td>1.0</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>Tam30</td>
<td>300–350</td>
<td>8,000–10,000</td>
<td>1.0</td>
<td>1.0</td>
<td>0.3</td>
</tr>
</tbody>
</table>

5.2 Results of unequal area fixed-shape FLPs

In this section, the FLPs of size 15, 20 and 30 departments from previous literature are attempted using BBO algorithm. Furthermore, two randomly generated large size FLPs having 40 and 50 departments are also attempted. The comparison of OFVs of 15, 20 and 30 department FLPs obtained using BBO algorithm with the results of previous researchers are presented in Table 2. The results obtained using BBO are found better than the previous researchers’ results. The percentage improvement in solutions is calculated using equation (16). It is observed that the performance of BBO is better compared to GA, HGA, TS and FA used by previous researchers (Table 2).

\[
\text{% Improvement} = \left( \frac{\text{Best known solution} - \text{Obtained solution}}{\text{Best known solution}} \right) \times 100
\]

**Table 2**  
**Comparison of results for unequal area fixed-shape FLPs**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 15</td>
<td>6,813</td>
<td>9,120</td>
<td>6,941.4</td>
<td>6,615.81</td>
<td>5,838.2</td>
<td>5,657.5</td>
<td>5.01</td>
</tr>
<tr>
<td>n = 20</td>
<td>13,190</td>
<td>21,885</td>
<td>14,696</td>
<td>13,198.40</td>
<td>11,505</td>
<td>10,952</td>
<td>4.81</td>
</tr>
<tr>
<td>n = 30</td>
<td>35,358</td>
<td>50,492</td>
<td>32,386</td>
<td>33,721.20</td>
<td>29,115</td>
<td>25,300</td>
<td>13.10</td>
</tr>
</tbody>
</table>
The comparison of computational time required for BBO and previously used algorithms is shown in Tables 3, 4 and 5. In Table 3, the CPU time (in terms of average and standard deviation) of BBO is compared with GA and HGA proposed by Lee and Lee (2002) and FA proposed by Ingole and Singh (2017). Table 4 and Table 5 shows the comparison of CPU time with HGA used by Schnecke and Vornberger (1997) and TS used by Scholz et al. (2009) respectively. It is observed that the computation time required by BBO is less than GA, HGA, FA and TS methods considered by previous researchers.

Table 3 CPU time compared with Lee and Lee (2002) and Ingole and Singh (2017)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P</th>
<th>G</th>
<th>Average/SD of CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n = 15</td>
<td>n = 20</td>
<td>n = 30</td>
</tr>
<tr>
<td>GA (Lee and Lee, 2002)</td>
<td>100–400</td>
<td>10</td>
<td>28.2/9.4</td>
</tr>
<tr>
<td>HGA (Lee and Lee, 2002)</td>
<td>100–400</td>
<td>10</td>
<td>34.1/11.9</td>
</tr>
<tr>
<td>FA (Ingole and Singh, 2017)</td>
<td>100–400</td>
<td>10</td>
<td>7.7/5.3</td>
</tr>
<tr>
<td>BBO (This paper)</td>
<td>100–400</td>
<td>10</td>
<td>6.8/4.7</td>
</tr>
</tbody>
</table>

Table 4 CPU time compared with Schnecke and Vornberger (1997) and Ingole and Singh (2017)

<table>
<thead>
<tr>
<th>Problem</th>
<th>P</th>
<th>G</th>
<th>CPU time/Run (seconds)</th>
<th>CPU time/generation (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>HGA (Schnecke and Vornberger, 1997)</td>
<td>FA (Ingole and Singh, 2017)</td>
</tr>
<tr>
<td>n = 15</td>
<td>40</td>
<td>2,000</td>
<td></td>
<td>81.21</td>
</tr>
<tr>
<td>n = 20</td>
<td>40</td>
<td>3,000</td>
<td>2,400</td>
<td>167.59</td>
</tr>
<tr>
<td>n = 30</td>
<td>40</td>
<td>3,000</td>
<td>6,000</td>
<td>284.01</td>
</tr>
</tbody>
</table>

Table 5 CPU time compared with Scholz et al. (2009) and Ingole and Singh (2017)

<table>
<thead>
<tr>
<th>Problem</th>
<th>P</th>
<th>G</th>
<th>CPU time/Run (seconds)</th>
<th>CPU time/generation (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>TS* (Scholz et al., 2009)</td>
<td>FA (Ingole and Singh, 2017)</td>
</tr>
<tr>
<td>n = 15</td>
<td>10</td>
<td>145</td>
<td>3.12</td>
<td>0.131</td>
</tr>
<tr>
<td>n = 20</td>
<td>10</td>
<td>185</td>
<td>50</td>
<td>0.270</td>
</tr>
<tr>
<td>n = 30</td>
<td>10</td>
<td>119</td>
<td>95.4</td>
<td>1.336</td>
</tr>
</tbody>
</table>

Note: *TS does not require population

The computation time for 15 department problem is not mentioned by Schnecke and Vornberger (1997) using HGA (Table 4). Scholz et al. (2009) used TS algorithm, a neighbourhood search algorithm which does not require population (Table 5), however, the CPU time of BBO is compared considering population size as 10 as Ingole and Singh (2017) have used the same population size to solve FLPs using FA.

Computation time is one of the important aspects while solving large size FLPs as they need larger time to reach at optimal solution. It is observed that the time required to obtain optimum result using BBO is less compared to the time taken by the algorithms used by previous researchers`; however the computation time differs as each computer system has different configuration and processor. The control parameters of an algorithm
should be adjusted appropriately to reach at optimal solution in reasonable time. There are some drawbacks of previously proposed algorithms. HGA required more population size and generations. The TS is a point-based algorithm which can trap in local minimum value. Further, the computational time required by HGA and TS are more than the considered BBO algorithm.

**Figure 5**  Layout solution for n= 15 FLP (see online version for colours)

**Figure 6**  Layout solution for n= 20 FLP (see online version for colours)

**Figure 7**  Layout solution for n= 30 FLP (see online version for colours)
The optimal layout solutions for $n = 15$, $20$ and $30$ size fixed-shape FLPs obtained using BBO algorithm are shown in Figures 5, 6 and 7 respectively. The blank space left at the bottom of the layout can be utilised for further facility integration in future.

Figure 8  Convergence of BBO for $n = 15$ FLP (see online version for colours)

Figure 9  Convergence of BBO for $n = 20$ FLP (see online version for colours)
The convergence graph of BBO algorithm for OFV against generations is shown in Figures 8, 9 and 10 for 15, 20 and 30 departments FLPs respectively. It is observed from Figures 8 to 10 that the convergence speed of BBO algorithm is fast for unequal area fixed-shape FLPs to reach at optimal solution. A suitable combination of population size, generations and other parameters of an algorithm can yield better results.

5.2.1 Numerical efficiency of BBO algorithm for large size FLPs

The efficiency of BBO algorithm is tested by solving large size FLPs presented by Ingole and Singh (2017). The problems consist of 40 and 50 departments. The parameters of BBO are kept same as FA for making fair comparison. The four sets of population and generations were considered by Ingole and Singh (2017). Each set runs for ten trials and the results are presented in Table 6. It can be observed from the comparison result (Table 6) that for every set, the considered BBO is giving better solution than FA. The improvement obtained in optimal solutions for 40 and 50 departments FLPs using BBO algorithm is 15.01% and 6.07% respectively. The improvement in the solution reveals that BBO is capable of providing optimal solutions to complex combinatorial optimisation problems. The computational time of BBO is also compared with FA and presented in Table 7. The required time is less or same as that of FA which proves the efficiency of BBO algorithm.
### Table 6
Results and comparison of $n = 40, 50$ using BBO

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>$P = 100$</th>
<th>$G = 500$</th>
<th>$P = 100$</th>
<th>$G = 1,000$</th>
<th>$P = 200$</th>
<th>$G = 500$</th>
<th>$P = 200$</th>
<th>$G = 1,000$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Best</td>
<td>Average</td>
<td>Best</td>
<td>Average</td>
<td>Best</td>
<td>Average</td>
<td>Best</td>
<td>Average</td>
</tr>
<tr>
<td>$n = 40$</td>
<td>FA</td>
<td>488,611</td>
<td>499,546.3</td>
<td>486,505</td>
<td>498,544.2</td>
<td>485,736</td>
<td>497,569.8</td>
<td>485,150</td>
<td>495,940.1</td>
</tr>
<tr>
<td></td>
<td>(Ingole and Singh, 2017)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BBO</td>
<td>430,839</td>
<td>443,525.5</td>
<td>412,779</td>
<td>431,597.4</td>
<td>419,117</td>
<td>427,825.1</td>
<td>414,365</td>
<td>417,554.6</td>
</tr>
<tr>
<td>$n = 50$</td>
<td>FA</td>
<td>2,056,984</td>
<td>2,084,621.9</td>
<td>2,036,950</td>
<td>2,076,700.2</td>
<td>2,033,821</td>
<td>2,074,298.3</td>
<td>1,995,866</td>
<td>2,046,281.0</td>
</tr>
<tr>
<td></td>
<td>(Ingole and Singh, 2017)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BBO</td>
<td>1,885,366</td>
<td>1,954,885</td>
<td>1,881,905</td>
<td>1,932,491</td>
<td>1,885,148</td>
<td>1,916,076</td>
<td>1,874,740</td>
<td>1,922,378</td>
</tr>
<tr>
<td>Improvement</td>
<td>$n = 40$</td>
<td>11.82%</td>
<td>15.15%</td>
<td>13.71%</td>
<td>14.59%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$n = 50$</td>
<td>8.34%</td>
<td>7.61%</td>
<td>7.31%</td>
<td>6.07%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 7  Computational time of BBO for n = 40, 50 size FLPs (minutes)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>Average time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 40</td>
<td>FA (Ingole and Singh, 2017)</td>
<td>3.3 6.6 8.3 13.3</td>
</tr>
<tr>
<td></td>
<td>BBO</td>
<td>3.1 5.9 8.2 13.6</td>
</tr>
<tr>
<td>n = 50</td>
<td>FA (Ingole and Singh, 2017)</td>
<td>5.0 10.8 12.3 24.9</td>
</tr>
<tr>
<td></td>
<td>BBO</td>
<td>4.8 10.1 12.4 23.7</td>
</tr>
</tbody>
</table>

The optimal layouts obtained for 40 and 50 department FLPs are presented in Figure 11 and Figure 12 respectively. The convergence graphs of BBO for these FLPs are shown in Figure 13 and Figure 14 respectively. From the convergence nature, it can be seen that BBO gradually proceeds towards near best solution in minimum number of generations.

Figure 11  Best layout for n = 40 FLP (see online version for colours)

![Figure 11](image1)

Figure 12  Best layout for n = 50 FLP (see online version for colours)

![Figure 12](image2)
5.3 Results of unequal area flexible shape FLPs

In this section, unequal area flexible shape FLPs are attempted using BBO algorithm. The datasets of test problems considered are presented in Table 8. The problems considered are most studied benchmark problems in the literature. Many researchers attempted these
problems using numerous optimisation techniques. The data regarding the height and width of layouts, the shape constraint and the source references are given in Table 8. The shape constraints are considered from Ulutas and Kulturel-Konak (2012). The areas of the facilities and flow matrices of all the problems are considered from the source references mentioned in Table 8.

The results of MHC of unequal area FLPs with FBS using BBO algorithm are presented in Table 9. BBO algorithm runs for 10 trials for each problem and average of the solutions are shown in Table 9 along with the computational time (CPU time) required to run one trial. It is seen from Table 9, that the computational time required to run BBO algorithm for small size problems is suitably low, however; the time required for large size FLPs is slightly more because the population and generations are higher. The layout solution strings are presented in Table 9. The partition line between the facilities denotes a new bay. Table 10 presents the comparison of the results obtained in this study with previous best known results.

Table 8  Datasets of unequal area flexible-shape FLPs

<table>
<thead>
<tr>
<th>Sr. no.</th>
<th>Problem</th>
<th>No. of facilities</th>
<th>Layout size</th>
<th>Shape constraint</th>
<th>Source reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>O7</td>
<td>7</td>
<td>13</td>
<td>(\alpha_{\text{max}} = 4)</td>
<td>Meller et al. (1999)</td>
</tr>
<tr>
<td>2</td>
<td>O8</td>
<td>8</td>
<td>13</td>
<td>(\alpha_{\text{max}} = 4)</td>
<td>Meller et al. (1999)</td>
</tr>
<tr>
<td>3</td>
<td>O9</td>
<td>9</td>
<td>13</td>
<td>(\alpha_{\text{max}} = 4)</td>
<td>Meller et al. (1999)</td>
</tr>
<tr>
<td>4</td>
<td>VC10Ea</td>
<td>10</td>
<td>25</td>
<td>(\alpha_{\text{max}} = 5)</td>
<td>Van Camp et al. (1992)</td>
</tr>
<tr>
<td>5</td>
<td>VC10Es</td>
<td>10</td>
<td>25</td>
<td>(l_{\text{min}} = 5)</td>
<td>Van Camp et al. (1992)</td>
</tr>
<tr>
<td>6</td>
<td>VC10Ra</td>
<td>10</td>
<td>25</td>
<td>(\alpha_{\text{min}} = 5)</td>
<td>Van Camp et al. (1992)</td>
</tr>
<tr>
<td>7</td>
<td>VC10Rs</td>
<td>10</td>
<td>25</td>
<td>(l_{\text{min}} = 5)</td>
<td>Van Camp et al. (1992)</td>
</tr>
<tr>
<td>8</td>
<td>BA12</td>
<td>12</td>
<td>6</td>
<td>(l_{\text{min}} = 1)</td>
<td>Bazaraa (1975)</td>
</tr>
<tr>
<td>9</td>
<td>BA14TS</td>
<td>14</td>
<td>7</td>
<td>(l_{\text{min}} = 1)</td>
<td>Tate and Smith (1995)</td>
</tr>
<tr>
<td>10</td>
<td>Tam20</td>
<td>20</td>
<td>35</td>
<td>(\alpha_{\text{min}} = 5)</td>
<td>Tam (1992) and Gau and Meller (1999)</td>
</tr>
<tr>
<td>11</td>
<td>Tam30</td>
<td>30</td>
<td>40</td>
<td>(\alpha_{\text{min}} = 5)</td>
<td>Tam (1992) and Gau and Meller (1999)</td>
</tr>
</tbody>
</table>

The comparison of the results obtained using BBO with previous best known results is presented in Table 10. It is to be specified that the comparison is made for the FLPs solved only using FBS in literature. Although the considered unequal area flexible shape FLPs are solved by many researchers such as Komarudin and Wong (2010), Anjos and Vieira (2016), Goncalves and Resende (2015), Xiao et al. (2017) and Kang and Chae (2017), however; these researchers solved the problems using different layout presentation methods instead of FBS, therefore the comparison is not appropriate with them.
The solutions of FLPs ‘O7’, ‘O8’ and ‘O9’ are compared with solutions obtained by Palomo-Romero et al. (2017) using IMGA. The results obtained for ‘O7’ and ‘O8’ problems using BBO are optimal and same as that of best known results. For ‘O9’ problem, the solution obtained using BBO is not better than the solution obtained by Palomo-Romero et al. (2017); however after verifying the layout solution of O9 problem given by Palomo-Romero et al. (2017), it is observed that the solution is not equivalent to the layout. In VC10Ea problem, the improvement in the solution is 0.24% as compared to the solution obtained by Ulutas and Kulturel-Konak (2012). The solution of VC10Es problem is not improved as compared to Kulturel-Konak and Konak (2011a) and Palomo-Romero et al. (2017); however it is improved as compared to the result obtained using PSO implemented by Kulturel-Konak and Konak (2011b) and CSA implemented by Ulutas and Kulturel-Konak (2012). The improvement in the solutions of VC10Ra, VC10Rs, BA12, BA14TS, Tam20 and Tam30 is found to be 1.55%, 3.21%, 0.02%, 0.92%, 0.10% and 0.13% respectively. BBO algorithm is successfully implemented on unequal area FLPs which are combinatorial optimisation problems and performing better than other well-known optimisation algorithms.

The optimal layouts obtained for ‘O7’, ‘O8’ and ‘O9’ are presented in Figure 15; for ‘VC10Ea’, ‘VC10Es’, ‘VC10Ra’ and ‘VC10Rs’ problems are presented in Figure 16; for ‘BA12’ and ‘BA14TS’ are presented in Figure 17 and for ‘Tam20’ and ‘Tam30’ are presented in Figure 18. In ‘O7’, ‘O8’, ‘O9’, ‘VC10’ and ‘BA14TS’, total layout area \((H\times W)\) is equal to the total area of all the facilities, therefore the entire floor area is occupied by the facilities, whereas; the layout areas of ‘BA12’, ‘Tam20’ and ‘Tam30’ are more than the total area of the facilities. Therefore, the unoccupied space is shaded in their respective layouts. This unoccupied space (also known as total blank area) can be used for future expansion of facilities.

The convergence graphs for flexible-shape FLPs are presented in Figure 19 (‘O7’, ‘O8’ and ‘O9’), Figure 20 (‘VC10Ea’, ‘VC10Es’, ‘VC10Ra’ and ‘VC10Rs’), Figure 21 (‘BA12’ and ‘BA14TS’) and Figure 22 (‘Tam20’ and ‘Tam30’). The generations required to obtain optimal solutions using BBO algorithm are more than methods used by previous researchers, therefore this method needs longer time for execution of one run. However, from the graphs it is observed that the algorithm converges to optimal solution in less number of generations. In BBO algorithm, the migration and mutation operators need a little longer time to expand the search space for obtaining best results. Therefore, the computational time required to obtain best layout is reasonable, as facility layout design is not a time bound problem (Komarudin and Wong, 2010).

From convergence graphs, it is concluded that BBO provides better results for FLPs. The migration and mutation operators in proposed algorithm are working efficiently to converge the solution space of the problem to near optimality. In unequal area FLPs, while placing unequal area shape facilities, number of constraints have to be considered such as aspect ratio, minimum length, blank space between the facilities, etc. Therefore quantitative factors such as length and width of the facilities and floor-space, material flow, number of trips and shape of the facilities can be considered as most critical factors while designing a facility layout.
<table>
<thead>
<tr>
<th>Sr. no.</th>
<th>Test problem</th>
<th>Best solution</th>
<th>Average</th>
<th>CPU time</th>
<th>Layout solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>O7</td>
<td>134.19</td>
<td>135.20</td>
<td>22.13</td>
<td>1 4 6 2</td>
</tr>
<tr>
<td>2</td>
<td>O8</td>
<td>245.51</td>
<td>257.84</td>
<td>35.08</td>
<td>5 8 6 3</td>
</tr>
<tr>
<td>3</td>
<td>O9</td>
<td>254.38</td>
<td>260.27</td>
<td>52.32</td>
<td>3 7 8 1</td>
</tr>
<tr>
<td>4</td>
<td>VC10Ea</td>
<td>18.753.01</td>
<td>19.462.66</td>
<td>284.52</td>
<td>1 6 2 9 3</td>
</tr>
<tr>
<td>5</td>
<td>VC10Es</td>
<td>19.978.13</td>
<td>22.691.14</td>
<td>293.22</td>
<td>9 2 7 6 4</td>
</tr>
<tr>
<td>6</td>
<td>VC10Ra</td>
<td>19.830</td>
<td>20.293.00</td>
<td>290.05</td>
<td>1 6 2 7 9 4</td>
</tr>
<tr>
<td>7</td>
<td>VC10Rs</td>
<td>22.164.26</td>
<td>22.876.25</td>
<td>289.56</td>
<td>11 2 9 6 1</td>
</tr>
<tr>
<td>8</td>
<td>BA12</td>
<td>8.019.53</td>
<td>8.317.1</td>
<td>611.23</td>
<td>6 1 2</td>
</tr>
<tr>
<td>9</td>
<td>BA14TS</td>
<td>4.798.7</td>
<td>4.943.4</td>
<td>675.50</td>
<td>4 9 8 12 13</td>
</tr>
<tr>
<td>10</td>
<td>Tam20</td>
<td>8.744.6</td>
<td>9.011.4</td>
<td>722.28</td>
<td>10 1 5</td>
</tr>
<tr>
<td>11</td>
<td>Tam30</td>
<td>19.436.9</td>
<td>20.436.6</td>
<td>1.606.5</td>
<td>1 2 10</td>
</tr>
<tr>
<td>Sr. no.</td>
<td>References of best known solutions</td>
<td>O7</td>
<td>O8</td>
<td>O9</td>
<td>VC10Ea</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>GA (Tate and Smith, 1995)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>AS (Komarudin and Wong, 2010)</td>
<td>131.68</td>
<td>243.12</td>
<td>236.14</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>ACO (Kulturel-Konak and Konak, 2011a)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>PSO (Kulturel-Konak and Konak, 2011b)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>CSA (Ulutas and Kulturel-Konak, 2012)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>18,798.70</td>
</tr>
<tr>
<td>6</td>
<td>AMPL/CPLEX (Anjos and Vieira, 2016)</td>
<td>-</td>
<td>-</td>
<td>259.00</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>IMGA (Palomo-Romero et al., 2017)</td>
<td>134.19</td>
<td>245.51</td>
<td>241.06</td>
<td>18,554.59</td>
</tr>
<tr>
<td>8</td>
<td>BBO algorithm</td>
<td>134.19</td>
<td>245.51</td>
<td>254.38</td>
<td>18,753.01</td>
</tr>
</tbody>
</table>

% Difference in the solution

|              | 0    | 0    | -    | 0.24 | -    | 1.55  | 3.21  | 0.02  | 0.92  | 0.10  | 0.13  |
Figure 15  
Best layouts of O7, O8 and O9

![Layout of O7](image1)
![Layout of O8](image2)
![Layout of O9](image3)

Figure 16  
Best layouts of VC10Ea, VC10Es, VC10Rb, VC10Rs

![Layout of VC10Ea](image4)
![Layout of VC10Es](image5)
![Layout of VC10Ra](image6)
![Layout of VC10Rs](image7)

Figure 17  
Best layouts of BA12 and BA14TS

![Layout of BA12](image8)
![Layout of BA14TS](image9)
Figure 18  Best layouts of Tam20 and Tam30 (see online version for colours)

Figure 19  Convergence of BBO for O7, O8 and O9 FLPs (see online version for colours)

Figure 20  Convergence of BBO for VC10 FLPs (see online version for colours)
6 Conclusions

Unequal area FLPs are the most studied problems by many researchers. This paper mainly focuses on unequal area FLPs having fixed-shape and flexible-shape facilities. In most of the manufacturing industries, the layout consists of unequal shape facilities/machines. Unequal area, fixed-shape FLPs having 15, 20 and 30 departments/facilities are optimised using BBO algorithm and the better results are obtained comparatively with respect to previous researchers’ results. Furthermore, two large size pseudo problems having 40 and 50 departments/facilities are also attempted to show the applicability of BBO algorithm to various size unequal area FLPs. The improvement in the solutions are obtained as 5.01%, 4.81% and 13.10% for 15, 20 and 30 departments FLPs respectively. For 40 and 50 departments FLPs the improvement is obtained using BBO algorithm is 15.01% and 6.07% respectively as compared to previous researchers’ results. BBO algorithm is further implemented to get the optimum layouts for unequal area, flexible-shape FLPs. Different size problems having 7, 8, 9, 10 (4 FLPs having 10 departments), 12, 14, 20 and 30 departments/facilities are attempted and the results are compared. It is observed that BBO is providing the better results than previous researchers’ results in less computation time. The results obtained for ‘O7’, ‘O8’ and ‘O9’ using BBO are optimal and found same as that of best known solutions. In
VC10Ea, using BBO 0.24% improvement is obtained compared to the result of CSA used by previous researchers. The improvement in the solutions of VC10Ra, VC10Rs, BA12, BA14TS, Tam20 and Tam30 FLPs obtained using BBO are 1.55%, 3.21%, 0.02%, 0.92%, 0.10% and 0.13% respectively compared to the previous researchers’ results.

A detailed demonstration of BBO algorithm to solve FLPs is provided for understanding the implementation of it in FLPs. The theoretical and managerial implications are that the solutions of the attempted FLPs obtained using BBO algorithm will be useful for comprehending more such type of FLPs and the management departments of the industries can think for improving or optimising the layout cost by doing further research from the solutions obtained and examine the optimisation technique utilised in the present work.

6.1 Limitations and future directions

The FLPs attempted in the present paper have considered only quantitative factors. More realistic layouts are possible if both quantitative as well as qualitative factors are considered simultaneously. The FLPs can be attempted using other newly developed and more recent algorithms to get the better layouts. BBO algorithm can bring future scope for researchers to solve dynamic and multi-objective FLPs which is very essential in today’s manufacturing scenario.

References


S. Ingole and D. Singh


Fixed and flexible shape facility layout problems


**Appendix**

*Demonstration steps of BBO algorithm with an illustrative example of n=5 FLP with FBS*

Step 1 Randomly generate islands ($s = 5$). Each island is considered as a layout. The encoding of FBS for unequal area FLP is considered from Ulutas and Kulturel-Konak (2012). Since there are 5 departments; string length of each island consists of $(2n–1)$, i.e., nine random numbers.

\[
\{0.8147 \ 0.0975 \ 0.1576 \ 0.1419 \ 0.6557 \ 0.7577 \ 0.7060 \ 0.8235 \ 0.4387\}
\]

\[
\{0.9058 \ 0.2785 \ 0.9706 \ 0.4218 \ 0.0357 \ 0.7431 \ 0.0318 \ 0.6948 \ 0.3816\}
\]

\[
\{0.1270 \ 0.5469 \ 0.9572 \ 0.9157 \ 0.8491 \ 0.3922 \ 0.2769 \ 0.3171 \ 0.7655\}
\]

\[
\{0.9134 \ 0.9575 \ 0.4854 \ 0.7922 \ 0.9340 \ 0.6555 \ 0.0462 \ 0.9502 \ 0.7952\}
\]

\[
\{0.0344 \ 0.6324 \ 0.0971 \ 0.9649 \ 0.8003 \ 0.9595 \ 0.6787 \ 0.1712 \ 0.1869\}
\]

Step 2 Obtain the integer numbers by sorting and indexing each row of above matrix to obtain the layout sequence.

$L_1: 2-4-3-9-5-7-6-1-8$

$L_2: 7-5-2-9-4-8-6-1-3$

$L_3: 1-7-8-6-2-9-5-4-3$

$L_4: 7-3-6-4-9-1-5-8-2$

$L_5: 1-3-8-9-2-7-5-6-4$

To get the locations of bay, replace the numbers above 5 with 0. The positions of bays are represented with 0. The layouts and the OFVs are evaluated.
From initial layouts, the minimum OFV is calculated as 539 and the corresponding layout is the best layout or best island.

Step 3  Start first iteration/generation. The emigration rate $\mu_s$ and immigration rate $\lambda_s$ are calculated using equations (1) and (2) respectively.

$$\mu_s = \{0.2, 0.4, 0.6, 0.8, 1\}$$
$$\lambda_s = \{0.8, 0.6, 0.4, 0.2, 0\}$$

Probability of the species $P_s$ is calculated using equation (4)

$$P_s = \{1, 5, 10, 10, 5, 1\}$$

Step 4  Start migration operation. Select two islands for emigration and immigration. Using $\mu_s$, third island is selected for emigration and using $\lambda_s$, fourth island is selected for immigration. Randomly select SIVs from third and fourth island to replace their positions. Suppose seventh and second SIVs are selected from third and fourth island respectively. Replace their positions as shown in bold case. The new islands obtained as follows:

\[
\begin{align*}
L_1 : & 2 - 4 - 3 - 0 - 5 - 0 - 0 - 1 - 0; & \text{OFV : } & 586 \\
L_2 : & 0 - 5 - 2 - 0 - 4 - 0 - 0 - 1 - 3; & \text{OFV : } & 638 \\
L_3 : & 1 - 0 - 0 - 0 - 2 - 0 - 5 - 4 - 3; & \text{OFV : } & 539 \\
L_4 : & 0 - 3 - 0 - 4 - 0 - 1 - 5 - 0 - 2; & \text{OFV : } & 690 \\
L_5 : & 1 - 3 - 0 - 0 - 2 - 0 - 5 - 0 - 4; & \text{OFV : } & 581 \\
\end{align*}
\]

**Step 5  Start mutation operation.** The mutation rate is calculated using equation (7).

$$m = \{0.8998, 0.4998, 0, 0, 0.4998, 0.8998\}$$

Two islands are selected using $P_s$. In each island, one SIV is selected to replace with randomly generated number. Suppose first and fifth islands are selected for mutation. From first island, third SIV is selected and from fifth island, ninth SIV is selected to replace with random numbers $r_1 = 0.2781$ and $r_2 = 0.3024$ respectively. After mutation the islands obtained are

\[
\begin{align*}
L_1 : & 0.8975, 0.1576, 0.1419, 0.6557, 0.7577, 0.7060, 0.8235, 0.4387 \\
L_2 : & 0.9058, 0.2785, 0.9706, 0.4218, 0.0357, 0.7431, 0.0318, 0.6948, 0.3816 \\
L_3 : & 0.1270, 0.5469, 0.9572, 0.9157, 0.8491, 0.3922, 0.9575, 0.3171, 0.7655 \\
L_4 : & 0.934, 0.2769, 0.4854, 0.7922, 0.9340, 0.6555, 0.0462, 0.9502, 0.7952 \\
L_5 : & 0.0344, 0.6324, 0.0971, 0.9649, 0.8003, 0.9595, 0.6787, 0.1712, 0.1869 \\
\end{align*}
\]
Find the integer numbers by sorting and indexing each row of above matrix to get the layouts and calculate their OFVs after first iteration.

1L : 2–4–3–0–5–0–0–1–0; OFV: 5 8 6
2L : 0–5–2–0–4–0–0–1–3; OFV: 6 3 8
3L : 1–0–0–2–0–5–4–3–0; OFV: 5 3 9
4L : 0–2–3–0–4–0–1–5–0; OFV: 6 1 2
5L : 1–3–0–0–2–0–5–0–4; OFV: 5 8 1

Step 7 After first iteration, the best layout and OFV obtained is:
1–0–0–2–0–5–4–3–0; OFV = 539

Step 8 Repeat steps 3 to 7 until maximum iterations are reached.