 ISSN 2008-0050 (Print), ISSN 1927-0097 (Online)

# Applying an immune ant colony system algorithm to solve an integrated flexible bay facility layout problem with input/output points design

Yun-Zhu Lin [1] and Yu-Chin Lin [1]
[1] Department of Civil Engineering, Chung Yuan Christian University, Taoyuan City, Taiwan
{mschang, yuchi}@cycu.edu.tw

Proc. ICAOR 2015
Vienna, Austria

*Keywords*:
Ant colony system
Clonal selection algorithm
Contour distance
Input/output (I/O) points design
Unequal-area facility layout

**Abstract**

Traditionally, most studies use a two-stage approach to solve a block layout problem with input/output points design. To make the planning results more practical and measure distance between facilities more precisely, this study integrates a flexible bay facility layout problem and input/output points design using a contour distance metric. In this study, an ant colony system (ACS), clonal selection algorithm (CSA) and shortest path algorithm are combined, and an immune ant colony system (IACS) algorithm is proposed to solve an unequal-area facility layout problem with input/output points design. Operations of CSA are embedded in the ACS to improve the solution quality of initial ant solutions and increase the differences between the ant solutions, and the search capability of IACS is thus enhanced. Nine international benchmark problems are used to test the algorithm efficiency of IACS. When compared to previous research, IACS can deliver new and better solutions.

## Introduction

Optimal layout design is an important topic in the early stage of new manufacturing systems. The Facility Layout Problem (FLP) is known as to have a substantial impact on operating costs, work in process, lead time and productivity. FLP is thus an essential optimization problem for many manufacturing and service organizations. The most common objective for facility design is to minimize total transportation costs (TTC), defined as the weighted sum of material flows between two departments, where weights are the material transportation distances along the flow paths (aisles) from output (pickup) points of a department to input (drop-off) points of another department. Allocating locations of departments within a facility and positioning input and output (I/O) points of departments are two interdependent design issues for FLP decision-making. Traditionally, they have been resolved separately in a sequential way because of the computational intractability of an integrated facility design problem. Based on rectilinear centroid-to-centroid distances, a block layout plan is generated, and then the locations of the I/O points are determined. In consideration of the need for measuring the contour distances between the I/O points along the perimeters of departmental boundaries, a block layout and the locations of I/O points should be simultaneously determined for better applicability in actual practice.

## Literature Review

Relatively speaking, the integrated design procedure of FLP has rarely been considered in the literature. Chittratanawat and Noble (1999) proposed an integrated approach for determining an equal-area block layout, I/O point locations, and a material handling equipment selection problem simultaneously. A tabu search metaheuristic procedure was used to solve the integrated problem which was formulated as a nonlinear mixed integer program. However, this same study is not applicable in real layout design projects because of the assumption of equal area in all departments. In addition, Norman *et al*. (2001b) defined three distinct intra-departmental flow types—U-shaped, linear, and C-shaped—and illustrated a discrete set of candidate I/O locations for each flow type.

Intra-departmental flows that occur within departments were considered for the FLPs. Due to the contour measure, the set of candidate I/O points for a individual department can be limited to the locations where that department intersects the corner of any if its adjacent departments. Considering such intra-departmental flow types, Shebanie II (2004) developed an integrated methodology that incorporated a genetic algorithm (GA) and a constructive heuristic to simultaneously solve the block layout problem of locating and shaping departments and the detailed design problem of locating the input/output

*Copyright © ORLab Analytics Inc. All rights reserved.*
*www.orlabanalytics.ca*

stations of different departments. A contour distance metric was used to evaluate the costs associated with material movement between the input/output stations of departments. A constructive placement heuristic developed to place the input/output stations was implemented as a subroutine to the genetic algorithm. For practical concerns, only one I/O point of each department is allowed to make the layout structure and the total cost consideration more realistic in certain cases.

For both situations, restricting the placement of I/O locations to one input station and one output station per department, Arapoglu *et al*. (2001) presented a two-phase, nested GA. Through mutation and crossover, a sequence of layouts was generated in the outer GA. Each layout was evaluated quickly using a constructive heuristic and a perturbation improvement algorithm in the inner GA. For these cases, the allowable number of I/O points per department is not constrained. Norman *et al*. (2001a) then adopted a flexible bay representation and presented a heuristic algorithm for determining the locations of I/O points embedded with the GA that was developed by Tate and Smith (1995) to obtain an optimal block layout with I/O point design based on contour distance merit. Conversely, Kim and Goetschalckx (2005) developed a method for determining the block layout, the locations of departmental I/O points, and the material flow paths between those I/O points. The topology of block layouts was represented by two linear sequences, which allows the layout to have either a slicing or a non-slicing structure. The block layout was obtained from the sequence-pair using a linear programming formulation while also using the rectilinear distance metric. A simulated annealing (SA) algorithm embedded with a linear programming algorithm, a shortest path algorithm, and I/O point location heuristics was developed to find the layout design using a low total transportation distance and the contour distances between the I/O points.

A FLP is a well-studied, combinatorial optimization problem. It is known to be complex and generally NP-Hard. Integrating more than two components of three layout components (block layout problem, I/O point location problem, and flow path design problem) in an integrated design procedure has rarely been considered in the literature because of computational difficulties. Due to this computational intractability, the majority of the research on FLPs has focused on heuristic approaches in order to find good solutions. Metaheuristic approaches, such as tabu search, GA, and SA, have been previously applied to these integrated FLPs. Applying the ant system, ant colony optimization, particle swarm optimization, and an artificial immune system to solve these integrated FLPs has not been reported in the literature. Compared with tabu search, GA, and SA, these approaches have been shown to be very effective in discovering the previously known best solutions and producing notable improvements of the FLPs.

## Problem Descriptions

The objective of the integrated FLP is to minimize the TTC using the contour distances between the I/O points of different departments. The followings assumptions were made:

1. The dimensions are given for the single floor area on which the departments are placed.
2. The departments have different sizes and rectangular shapes with a requirement for aspect ratios. They are located in parallel bays with varying widths, that is, the flexible bay structure (FBS) partitions the floor area.
3. The space for the flow paths is not considered, but it is included in the areas of departments. The facility area equals the total sum of all the department areas.
4. The intra-departmental workflow is C-shaped (Norman *et al*., 2001b). Each department has one input/output point. Adjacent departments, however, can share the same I/O point.
5. Materials are moved along the boundaries of the departments, so the input/output points are located on the perimeter of each department except for the two boundaries of the floor area. When the direction of the bay break is horizontal, the I/O points aren't allowed to be located at the top and bottom boundaries. When the direction of the bay break is vertical, the I/O points aren't allowed to be located at the right and left boundaries.

Based on the last two assumptions above, the I/O points are located along the inner edges of the flexible bays. We considered eight candidate locations of I/O points for each department, as shown in the middle of Figure 1. One example having four candidates of I/O positions is shown in the left part of Figure 1. Points 1, 2, 3, and 4 represent the left-bottom, left-top, right-top, and right-bottom corners, respectively. There are 12 candidate locations for I/O points that are coded as (*n*-*m*). The first number *n* represents the department. The second number *m* represents the corner position. In order to minimize the TTC, the locations of I/O points can be chosen by selecting the set for the shortest paths between all pairs of input point sand output points that have material flows once the block layout is determined. If workflows between all the departments are positive, an all-to-all shortest flow path network can be correspondingly constructed, as shown in the red lines in the right part of Figure 1. Then the TTD can be obtained. Note that in this study that the same candidate location has several coding numbers for nodes. They will be regarded as different nodes when we perform shortest path searching.

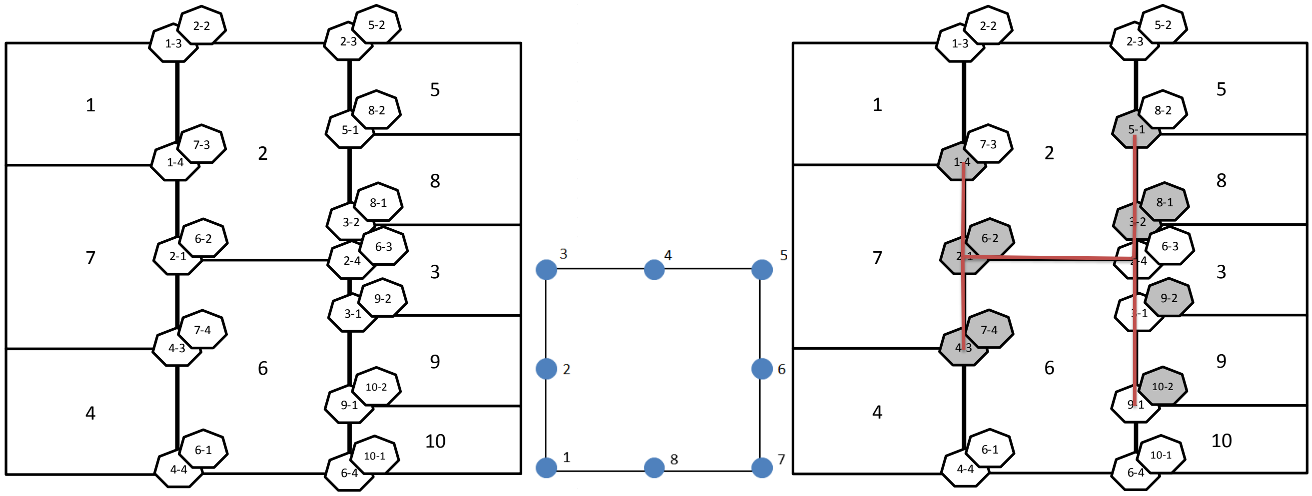


**Fig. 1.** Example of Integrated FLP

## Algorithms

Chang and Lin (2012) developed an ant colony system (ACS) algorithm with local search for solving the FLPs using a FBS representation. Compared with the previously best known solutions, ACS can obtain the same or better solutions for certain benchmark problems. Chang and Lin (2014) combined the clonal selection algorithm (CSA) and the previously developed ACS to propose an immune ant colony system (IACS) algorithm to solve the same benchmark problems. CSA is one of the population-based AIS algorithms. The researchers also pointed out that the convergence speed of the ACS can be improved by introducing CSA operations. Due to the computational intractability of the integrated FLP, we extend the IACS developed by Chang and Lin (2014) to solve this integrated flexible bay FLP. We embed the procedures for shortest flow path searching in IACS to calculate contour distances between the I/O points of departments for each ant solution. Thus, a flexible bay layout and the locations of I/O points are simultaneously determined. The solving procedures, called IACS_SPATH, are presented in Figure 2.

| | |
|---|---|
| **Initialization** | Step 0: Parameter setting and initialization<br>Step 0.1: parameters of ACS: maximum number of iterations (*NI*), number of ants (*N*), pheromone information parameter ($\alpha$), heuristic information parameter ($\beta$), and evaporation rate ($\rho$).<br>Step 0.2: parameters of CAS: size of memory pool, *r* = *N*×*b*%, clone number of best ant-solutions, $s_1 = (N-r)d\%$, and clone number of diverse ant-solutions $s_2 = (N-r)(1-d\%)$.<br>Step 0.4: Initialize pheromone information $\tau_{ij}^0, \forall i, j$.<br>Step 0.5: Initialize the fitness value for the global best solution. Set $z^* = \infty$.<br>Step 0.6: Generate an empty memory pool *M* |
| **Generation**<br>(Constructive Heuristics)<br>+ (Graph Algorithm) | Step 1: Generate initial candidate pool *P* of ant colony (2*N* ants)<br>Step 1.1: Perform the ***ant solutions construction procedure*** (Chang and Lin, 2012) to create 2*N* ants.<br>Step 1.2: Perform ***shortest flow path searching*** to determine the best I/O locations<br>Step 1.3: Evaluate the fitness of the ant colony in candidate pool *P* |
| **Clone** | Step 2: Generate a temporary pool *C* from the memory pool *M* and the candidate pool *P*<br>Step 2.1: Clone the ants in memory pool *M* (*r* ants) into the temporary pool *C*.<br>Step 2.2: Clone the best ants in candidate pool *P* ($s_1$ ants) into the temporary pool *C*.<br>Step 2.3: Evaluate the diversity measurement between each ant and the best ant in the candidate pool *P*, $\delta = \sum_l \left\| b_l - b_l^* \right\|$, where $b_l$ and $b_l^*$ is the current and the best bay width of bay *l*.<br>Step 2.4: Clone the diverse ants in candidate pool *P* ($s_2$ ants) into the temporary pool *C*. |
| **Mutation**<br>(Local Search) + **(**Graph Algorithm) | Step 3: Generate a mutated ants pool *C*1 from the temporary pool *C*<br>Step 3.1: Perform mutation operations for a department sequence and/or a bay break to all ants in temporary pool *C*.<br>Step 3.2: Perform ***shortest flow path searching*** to determine the best I/O locations<br>Step 3.3: Evaluate the fitness of all ants in the mutated ant pool *C*1 |
| **Replace** | Step 4 Update the temporary pool *C*<br>If the mutated ant is better than the original ant, replace the original ant. |

| | |
|---|---|
| **Selection** | Step 5: Select an ant colony (*n* ants) from temporary pool *C* |
| **Ant Colony Optimization** | Step 6: Perform an optimization search of the ant colony<br>Step 6.1: Exploit the selected regions by sending the ants on a local search by performing a state transition rule.<br>Step 6.2: Update the local pheromone for all the ants. |
| **Mutation** (Local Search) + (Graph Algorithm) + (Update Local Best) | Step 7: Mutate the current ant solutions for this iteration and perform local search operations to determine the current best solution<br>Step 7.1: Determine the threshold of the mutation rate, $\phi = N/\sum_p z_p$ .<br>Step 7.2: Calculate the mutation rate of ant *p*, $\phi_p = 1/z_p$ . If the value $\phi_p$ is less than the threshold $\phi$ , continue; otherwise, go to Step 8.<br>Step 7.3: Perform ***local search operations*** (Chang and Lin, 2012) for a department sequence and a bay break to ant solution *p*.<br>Step 7.4: Perform ***shortest flow path searching*** to determine the best I/O locations Step 7.5: Calculate the fitness value $\hat{z}_I$ of the ant *p* after a local search.<br>Step 7.6: Update the best solution for this iteration $z_I^*$ , once a new best solution is found, $\hat{z}_I < z_I^*$ .<br>Step 7.7: Update the local pheromone of the mutated ant *p*, if its fitness value is improved.<br>Step 7.8: Update the global pheromone of the mutated ant *p*, if its fitness value is not improved.<br>Step 7.9: Add ant *p* or mutated ant *p* to the mutated ants pool *C*1. |
| **Clone and Delete** | Step 8: Update the memory pool *M*<br>Step 8.1: Clone the best *r* ants in the mutated ants pool *C*1 into the memory pool *M*.<br>Step 8.2: Delete identical ants in memory pool *M* |
| **Delete and Replace** | Step 9: Update the candidate pool *P*<br>Step 9.1: Delete identical ants in candidate pool *P* to maintain ant diversity.<br>Step 9.2: Replace those ant solutions in the candidate pool *P* with rest ants with better fitness in the mutated ants pool *C*1. |
| **Update Global Best** | Step 10: Update the global best solution<br>If $z_I^*$ is less than $z^*$, update the fitness value of the global best solution $z^* := z_I^*$ . |
| **Stop Check** | Step 11: Stop criteria<br>If the maximum number of iterations is realized, then output the global best solution and stop; otherwise, go to Step 2. |

**Fig. 2.** Framework of the IACS Aapproach to Solve the FLP Using I/O Points Design

### *Solution Representation*

We adopt the ant solution representation as proposed by Komarudin (2009) for solving flexible bay facility layout. Each ant solution has two parts: Department sequence codes and bay break codes. The former represents the order of *n* departments that will be placed into the facility. The latter are *n* binary numbers. Here, 1 represents a bay break and 0 otherwise. Then, we consider the eight points of each department as candidate locations of the I/O points for each department. We also eliminate the dominated points from the set of candidate positions by checking for sufficient conditions (Kim & Kim, 1999).

### *Shortest Flow Path Searching*

At the first iteration, choose the first number of the department sequence code, i.e., Department *i*, as the origin of the shortest flow path. Choose the department with the maximal workflow between that department and Department *i*, i.e., *Department j*, as the destination for the shortest flow path. Perform Floyd's algorithm to find the all-to-all shortest paths along the department boundaries from the I/O point *r* of Department *i* to the I/O point *s* of Department *j*. We then can find the best I/O points for Department *i* and Department *j*.

$$\hat{r}_i = \arg\min TTC_{rs}^{ij} = \arg\min f_{ij}c_{ij} \sum_{a \in P_{irjs}} D_{ir \to js}\left((x_{ir}, y_{ir}), \ldots, (x_{js}, y_{js})\right) \tag{1}$$

$$\hat{s}_j = \arg\min TTC_{rs}^{ij} = \arg\min f_{ij}c_{ij} \sum_{a \in P_{irjs}} D_{ir \to js}\left((x_{ir}, y_{ir}), \ldots, (x_{js}, y_{js})\right) \tag{2}$$

where $a \in P_{irjs}$ represents all the links along the path between I/O point *r* and *s*; *D*(.) means a contour measure of distance; $(x_{ir}, y_{ir})$ represents the coordinate pair of I/O points.

At the following iterations, choose the best I/O point for the Destination *j,* as the origin of the shortest flow path, i.e. $r_j = \hat{s}_j$ . Choose the department with the maximal workflow between it and Destination *j*, i.e., Department *k*, as the destination of the shortest flow path. Perform Dijkstra's algorithm to find one-to-all shortest paths along the department boundaries from I/O point $r$ of Department $j$ to I/O point $s$ of Department $k$. We can find the best I/O point of the Department $k$. Continue searching the shortest flow path until all departments have been assigned an I/O point.

$$\hat{s}_k = \arg\min TTC_{rs}^{jk} = \arg\min f_{jk} c_{jk} \sum_{a \in P_{jrks}} D_{jr \to ks}\left(\left(x_{jr}, y_{jr}\right), \ldots, \left(x_{ks}, y_{ks}\right)\right) \tag{3}$$

***Evaluating the Fitness of the Solutions***

The fitness value of each ant solution is defined as the weighted total sum of TTC and penalty costs:

$$\begin{aligned} \min \quad z = TTC + \varpi P = &\sum_i \sum_j f_{ij} c_{ij} d_{rs}^{ij*} \\ &+ \varpi \sum_i \sum_j f_{ij} c_{ij} \left[ \max\left(0, Ub_i^w - w_i\right) + \max\left(0, Lb_i^w - w_i\right) + \max\left(0, Ub_i^h - h_i\right) + \max\left(0, Lb_i^h - h_i\right) \right] \end{aligned} \tag{4}$$

where $f_{ij}$ is the workflow from $i$ and $j$; $c_{ij}$ is the cost per unit distance from $i$ and $j$; $d_{rs,x}^{ij}$ is the contour distance of the I/O point $r$ from department $i$ to the I/O point $s$ of department $j$ on the $x$-axis; $d_{rs,y}^{ij}$ is the contour distance on the y-axis; and the minimal contour distance between these two departments is defined as: $d_{rs}^{ij*} = d_{rs,x}^{ij*} + d_{rs,y}^{ij*}$, and should be determined by shortest path searching. $\varpi$ is a weight of penalty costs; $Lb_i^h$ and $Lb_i^w$ are the lower height limit and the lower width limit of department $i$; $Ub_i^h$ and $Ub_i^w$ are the upper height limit and the lower width limit of department $i$; $h_i$ is the height of $i$; and $w_i$ is the width of $i$;. The penalty function was presented by Kulturel-Konak and Konak (2011).

## Numerical Examples

The proposed algorithm was tested using the problem sets listed in Table 1. All parameter values were determined based on previous research and the pre-tuning conducted by this study. The algorithm was replicated 10 times. We follow the Taguchi design of the experiment to determine the best parameter combination. We set a maximum number of iterations (*NI*), an evaporation rate ($\rho$), a probability for choosing a solution component ($q0$), the number of ants ($M$), a pheromone information parameter ($\alpha$), and a heuristic information parameter ($\beta$), equal to 35, 0.1, 0.5, 30, 5, and 3, respectively. For the first eight problems, new best FBS solutions were found the first time. The results of Problem Random10 were then compared to its best-known solutions. We found new best solutions that substantially improved the previous best-known solutions by 316.5%. Layout design results are shown in Figure 3.

**Table 1.** Comparison of the Best FBS Solutions with Other Approaches

| *Problem* | *Problem data* | *No. of department* | *Shape* | *Kim and Goetschalckx (2005)* | *This study* |
|---|---|---|---|---|---|
| O7 | Meller *et al*. (1998) | 7 | Aspect ratio ≤ 4 | - | **24.3** |
| O8 | Meller *et al*. (1998) | 8 | Aspect ratio ≤ 4 | - | **52.2** |
| O9 | Meller *et al*. (1998) | 9 | Aspect ratio ≤ 4 | - | **59.3** |
| vC10a | van Camp *et al*. (1991) | 10 | Minimal side ≥ 5 | 3230.7* | **4411.97** |
| M11 | Bozer *et al*. (1994) | 11 | Aspect ratio ≤ 5 | - | **282.34** |
| Ba12 | Bazaraa (1975) | 12 | Minimal side ≥ 1 | - | **4990.84** |
| Ba12TS | Bazaraa (1975) | 16 | Minimal side ≥ 1 | **-** | **5878.14** |
| Ba14 | Bazaraa (1975) | 14 | Minimal side ≥ 1 | - | **5259.38** |
| Random10 | Kim and Goetschalckx (2005) | 10 | Aspect ratio ≤ 4 | 976.19 | **234.37** |

* It is a slicing tree structured layout

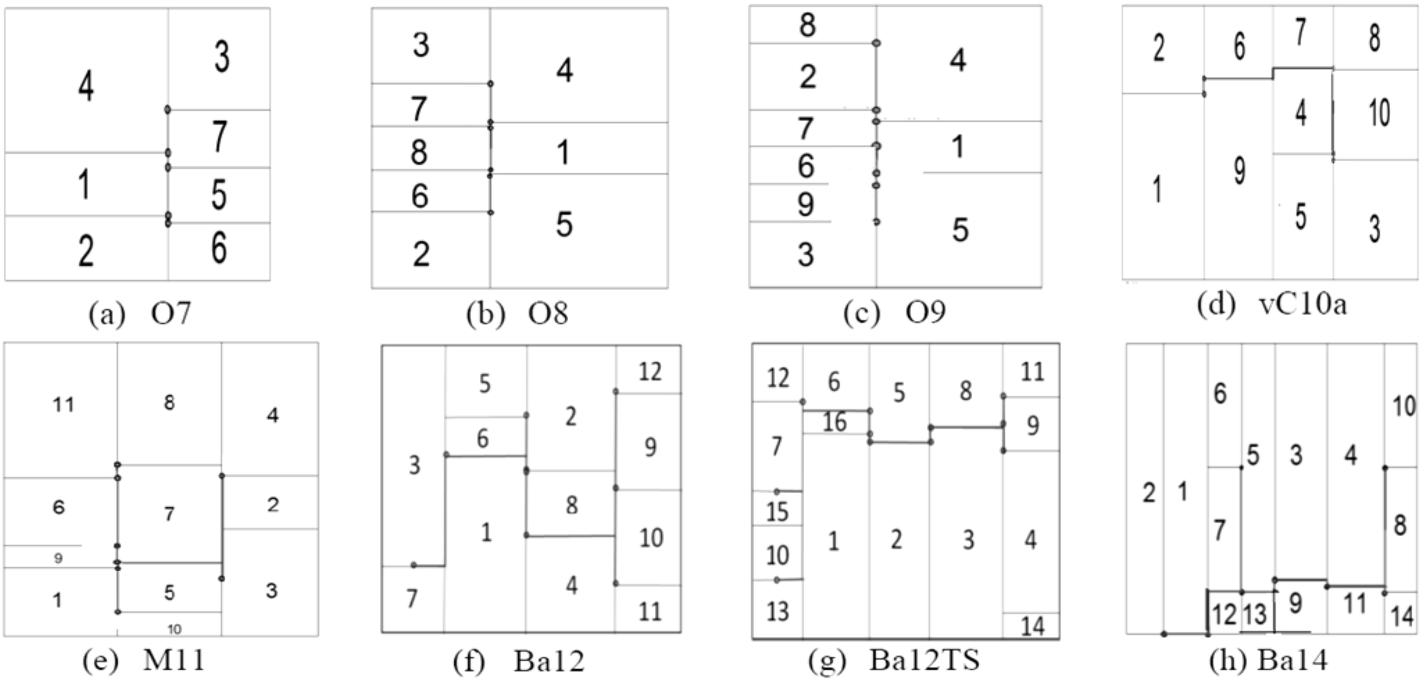


**Fig. 3.** Layout Design Results

## Conclusion

An IACS_SPATH algorithm is proposed in this study to simultaneously solve unequal-area FLP using I/O points design. By combining ACS with CSA and a shortest path searching procedure, IACS_SPATH can provide more efficient and comprehensive exploitation and exploration and also avoid stagnation at the local optima. The comparative results show that the IACS_SPATH approach is very promising. New best FBS solutions were found for certain international benchmark problems. This future work should include identifying more complicated local searches in order to achieve better results for medium and large instances. Moreover, the heuristic information function will be modified and improve the efficiency of the IACS. We also believe it would be interesting to consider limiting the number of I/O points.

## References

Arapoglu, R. A., Norman, B. A. and Smith, A. E. (2001). Locating input and output points in facilities design – a comparison of constructive, evolutionary, and exact methods. IIE Transactions on Evolutionary Computation, 5, 192-203.

Bazaraa, M. S. (1975). Computerized layout design: a branch and bound approach. AIIE Transactions, 7(4), 432-437.

Bozer, Y. A., Meller, R. D. and Erlebacher, S. J. (1994). An improvement-type layout algorithm for single and multiple floor facilities. Management Science, 40, 18-32.

Chang, M. S. and Lin, H. Y. (2012). A flexible bay structure representation and ant colony system for unequal area facility layout problems. Lecture Notes in Engineering and Computer Science, 2199(1), 1346-1351.

Chang, M. S. and Lin, H. Y. (2014). Applying an immune ant colony system algorithm to solve unequal area facility layout problem. Journal of Advanced Engineering, 9(3), 171-177.

Chittratanawat, S. and Noble, J. S. (1999). An integrated approach for facility layout, P/D locations and material handling system design. International Journal of Production Research, 37(3), 683-706.

Kim, J.-G. and Kim, Y.-D. (1999). A branch and bound algorithm for locating I/O points of departments on the block layout. Journal of Operational Research Society, 50, 517-525.

Kim, J.-G., and Goetschalckx, M. (2005). An integrated approach for the concurrent determination of the block layout and the input and output point locations based on the contour distance. International Journal of Production Research, 34, 2027-2047.

Komarudin (2009). An improved ant system algorithm unequal area facility layout problems, Master Thesis, University Technology Malaysia, Malaysia.

Kulturel-Konak, S. and Konak, A. (2011). Unequal area flexible bay facility layout using ant colony optimization. International Journal of Production Research, 49(7), 1877-1902.

Meller, R. D., Narayanan, V. and P. H. Vance, P. H. (1998). Optimal facility layout design. Operations Research Letters, 23, 117-127.

Norman, B. A., Arapoglu, R. A. and Smith, A. E. (2001a). Integrated facilities layout using a contour distance measure. IIE Transactions, 33, 337-344.

Norman, B.A., Smith, A. E., Yildirim, E. and Tharmmaphornphilas, W. (2001b). An evolutionary approach to incorporating intradepartmental low into facilities design, Advances in Engineering Software, 32, 443-453.

Shebanie II, C. R. (2004). An integrated, evolutionary approach to facility layout and detailed design. Master Thesis, University of Pittsburgh, USA.

Tate, D. M. and Smith, A. E. (1995). Unequal-area facility layout by genetic search. IIE Transactions, 27, 465-472.

van Camp, D. J., Carter, M. W. and Vannelli, A. (1991). A nonlinear optimization approach for solving facility layout problems. European Journal of Operational Research, 57, 174-189.