先進工程學刊 第九卷 第三期
Journal of Advanced Engineering Vol. 9, No. 3, pp. 169-175 / July 2014

應用免疫蟻群系統演算法求解部門大小不一致設施規畫問題
Applying an Immune Ant Colony System Algorithm to Solve Unequal Area Facility Layout Problem

張美香* 林欣怡
Mei-Shiang Chang*, Hsin-Yi Lin

摘要

本研究採用彈性區帶架構，結合克隆選擇演算法及蟻群系統演算法，提出一免疫蟻群系統演算法求解部門大小不一致設施規畫問題。克隆選擇演算法機制的導入可以改善蟻群系統演算法之收斂速度及增加蟻群解間的差異性，故可強化免疫蟻群系統演算法之搜尋能力。將免疫蟻群系統演算法應用於九種標竿問題求解，並與其他研究比較，證實本演算法的搜尋機制可以更快求得最佳的解答。

關鍵詞：蟻群系統，克隆選擇演算法，彈性區帶架構

Abstract

In this research, the clonal selection algorithm and an ant colony system are combined to propose an immune ant colony system algorithm to solve unequal-area facility layout problems using a flexible bay structure representation. Clonal selection algorithm operations are introduced in the ant colony system to improve the convergence speed of the ant colony system and increase the differences among ant solutions. The search capability of the immune ant colony system is thus enhanced. Datasets for well-known benchmark problems were used to evaluate the effectiveness of this approach. Compared with previous research efforts, the immune ant colony system can offer better solutions in a shorter timeframe for most benchmark problems.

Keywords: ant colony system, clonal selection algorithm, flexible bay structure

I. INTRODUCTION

Facility layout significant impacts the performance of a manufacturing or service industrial system since it can affect the flow of materials. Material movements are considered as a non-value added process usually. Therefore, facility layout should be a plan that controls needless material movement. Facility layout problems (FLPs) concerning space layout optimization have been investigated in depth by researchers in many fields, including manufacturing cell design, hospital design, and service center design. Given the number of departments, the area of each department, the cost and flow values associated with each pair of departments, unequal area FLPs seek to determine the optimal arrangement within each facility, so total material movement cost is minimized. Based on the unique requirements of each physical operation, each department needs to address certain ratio constraints or minimum length constraints to avoid narrow department shapes.

Many schemes that define how departments can be arranged within a facility area have been proposed to represent a FLP. Tong [1] first defined the flexible bay structure (FBS), a continuous layout representation. Rectangular-shaped departments are placed in parallel bays with varying widths. Bay boundaries form the basis of an aisle structure, and a bay structure can be regarded as a candidate for aisles. It helps with the transfer from a block layout design to an actual facility designed for practitioners [2]. In addition, the facility layout software packages, BLOCPLAN and SPIRAL, also can generate layouts similar to the FBS [3]. So we know that FBS has certain advantages to consider for an actual facility design [4] that make it one of the more practical layout representation schemes.

A FLP is a well-studied combinatorial optimization problem. FLPs are known to be complex and generally NP-Hard [5]. Exact approaches have only been able to obtain optimal solutions for small size instances up to 13 facilities [6]. Due to the computational intractability of a FLP, the majority of research on FLPs has focused on heuristic approaches to find good solutions. Metaheuristic approaches such as simulated annealing (SA) [7], genetic algorithms (GAs) [8-11], tabu search (TS) [12-15], ant

中原大學土木工程學系
*Corresponding author. E-mail: mschang@cycu.edu.tw
Department of Civil Engineering, Chung Yuan Christian University, Chung Li, Taiwan, R.O.C.
Manuscript received 10 February 2014; revised 27 May 2014; accepted 24 June 2014

system (AS) [16, 17], ant colony optimization (ACO) [18], particle swarm optimization (PSO) [3], and artificial immune system (AIS) [19] have been previously applied to FLPs. Compared with metaheuristics, such as GA, TS, AS, and exact methods, the ACO approach is shown to be very effective in discovering the previously known best solutions and producing notable improvements.

Recently biology-inspired AIS methods have been widely used for different optimization problem. AIS is a more recent branch of stochastic search algorithms and is classified as a population-based metaheuristic method. The clonal selection algorithm (CSA), the immune network algorithm (INA), and the negative selection algorithm (NSA) are three commonly applied types of AISs. CSA have been successfully applied to solve combinatorial optimization problems from various areas [20]. CSA is one of the population-based AIS algorithms. It was inspired by the clonal selection theory of acquired immunity that explains how basic natural immune improve their responses to the stimulation of non-self cells over time. This phenomenon is affinity maturation, that is, those antibodies that are capable of recognizing the antigens will proliferate.

AIS can be viewed as a swarm like system where there are many interacting agents that operate on multiple timescales that collectively maintain the host, through a process of collaboration and competition [21]. They also argue that AIS and swarm intelligence (SI) share many similarities, both at the methodological level and the algorithmic level. In 2010, Timmis *et al.* pointed out that these two approaches should be utilized in a complementary, rather than in a competitive, manner, to help solve complex engineering problems.

An ant system approach was first presented to solve the FLP in 2010. Wong and Komarudin [17] presented an improvement to the FBS representation by using free or empty space. Their algorithm can also improve the best known solution for several problem instances. Komarudin and Wong [16] used a slicing tree structure to represent FLP and integrated nine types of local search to improve algorithm performance. No doubt this heuristic shows encouraging results in solving FLPs. Chang and Lin [22] used an ant colony system (ACS) to solve the FLP with FBS. Compared to the previously best known solutions, an ACS can obtain the same or better solutions for some benchmark problems. Yet due to long calculation time and slow convergence speed, such an ACS cannot be used directly in FLPs for medium and large instances. These interesting researches inspire us to propose a novel metaheuristic, immunized ant colony system (IACS), to solve unequal area FLPs using FBS representation. By combining ACS with CSA, IACS can provide more efficient and comprehensive exploitation and exploration to improve the slow convergence and also avoid stagnation at the local optima.

The remainder of this paper is organized as follows: Section 2 offers a literature review of hybrid ant colony search to solve FLPs. Next in Section 3, a metaheuristic, which integrates ACS with CSA, is described. Then, experiments, results, and comparative study with other algorithms are presented and discussed in Section 4. Finally, Section 5 offers our conclusion with discussion and suggestions for future researches.

II. LITERATURE REVIEW

To our knowledge, many different hybrid methods have been offered to solve a FLP. Gambardella *et al.* [23] proposed the hybrid ant system HAS-QAP to solve a quadratic assignment problem (QAP). The originality of their approach is that the pheromone trail was not used to construct solutions, but rather to modify them in the local search. Talbi *et al.* [24] presented a parallel model for ACO using a tabu local search procedure to solve a FLP as a QAP. Exploration of the search space is guided by the evolution of pheromones levels, while exploitation was improved by a tabu local search heuristic. Testing results indicate that they compare favourably with other algorithms. Pour and Nosraty [25] solved a discrete FLP as a QAP by applying an ACO approach and a local search. Their testing results reveal that this metaheuristic can compete with other current solutions with encouraging results. Hani *et al.* [26] proposed a hybrid ACO approach coupled with a guided local search (GLS) to solve an industrial layout problem as a QAP. GLS uses an augmented cost function to guide the local search out of a local optimum. They emphasize that ACO-GLS is the most adaptable algorithm for this industrial case. Nourelfath *et al.* [27] used a hybrid approach that combines ACO with the extended great deluge local search technique to solve the discrete FLP as a QAP. The experimental results indicate that this metaheuristic offers advantages over other metaheuristics in terms of the quality of the solution.

III. THE IMMUNE ANT COLONY SYSTEM ALGORITHM

1. Solution Representation

We adopt the ant solution representation proposed by Komarudin [28] for solving FLPs. 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 is n binary numbers. Here, 1 represents a bay break, and 0 otherwise. We assume that bays run vertically and departments are placed from left to right and bottom to top. For example, a FLP with seven departments is shown in Fig. 1. The department sequence codes are 2-1-4-3-7-5-6. The bay break codes are 0-0-1-0-0-0. They mean that there are two bays. The first bay contains departments {2, 1, 4}. The second bay contains departments {3, 7, 5, 6}.

2. State Transition Rule and Heuristic Information Definition

The state transition rule is given by Eqs. (1) and (2), which shows how the ant k in department i chooses the j-th position of the department sequence to move to

$$j=\begin{cases}\arg\max\left\{\left[\tau_{ij}\right]^{\alpha}\cdot\left[\eta_{ij}\right]^{\beta}\right\}, \text{ if } q\le q_0 & \text{(exploitation)}\\ S & \text{,otherwise (exploration)}\end{cases} \quad (1)$$

$$S=P_{ij}^{k}=\begin{cases}\left[\tau_{ij}\right]^{\alpha}\cdot\left[\eta_{ij}\right]^{\beta}\Big/\sum_{q\in N_i}\left[\tau_{iq}\right]^{\alpha}\cdot\left[\eta_{iq}\right]^{\beta}, & \text{if } j\in N_i\\ 0 & \text{,otherwise}\end{cases} \quad (2)$$

where s is a probability to locate department j after department i in the positioning order of departments; τ_{ij} is the pheromone value defined as the relative desirability of assigning department j after department i in the department sequence; η_{ij} is the heuristic information related to assigning department j after department i in the department sequence; q is a random number uniformly distributed in [0, 1]; q_0 is a fixed parameter ($0 \le q_0 \le 1$); α is a parameter that determines the relative weight of pheromone information; and β is a parameter that determines the relative weight of heuristic information; P_{ij}^{k} is a probability of the department j of the department sequence to be chosen by an ant k located in department i and N_i is the available alternatives for the department sequence to be chosen by the corresponding ant located in department i.

Komarudin [28] offered this intuitive rule: "A department with higher material flow should be located nearer to the center of the facility." This heuristic information function is defined by Eq. (3).

$$\eta_{ij}=\left(\sum_{k=1}^{N}f_{ki}+\sum_{k=1}^{N}f_{ik}\right)\left(\frac{W}{2}-\left|x_j-\frac{W}{2}\right|+\frac{H}{2}-\left|y_j-\frac{H}{2}\right|\right) \quad (3)$$

where f_{ij} is the workflow from i and j; x_j is the x-coordinate of the centroid of the department j; and y_j is the y-coordinate of the centroid of the department j. The rectilinear distance between the centroid of the candidate department and the facility boundary is measured.

3. Pheromone Updating Rules

Local pheromone updating rule is as follows:

$$\tau_{ij}:=(1-\rho)\cdot\tau_{ij}+\rho\cdot\tau_0 \quad (4)$$

where $0 < \rho < 1$ is the evaporation parameter; τ_0 represents the initial level of the pheromone.

Fig. 1 A layout example using a FBS representation

Update the global pheromone of the mutated ant p according to Eqs. (5) and (6).

$$\tau_{ij}:=(1-\rho)\cdot\tau_{ij}+\rho\cdot\Delta\tau_{p\in P} \quad (5)$$

$$\Delta\tau_{p\in P}=\sum_{ij\in P}d_{ij}^{x}+d_{ij}^{y} \quad (6)$$

4. Evaluating the Fitness of Solutions

An ant solution s is feasible if its aspect ratio is less than the maximum allowable aspect ratio. Infeasible solutions are penalized by a penalty function as follows:

$$\begin{aligned}z=&\sum_i\sum_j f_{ij}c_{ij}\left(d_{ij}^{x}+d_{ij}^{y}\right)+\lambda\sum_i\left[Ub_i^{w}-w_i\right]^{+}+\left[Lb_i^{w}-w_i\right]^{+}\\&+\lambda\sum_i\left[Ub_i^{h}-h_i\right]^{+}+\left[Lb_i^{h}-h_i\right]^{+}\end{aligned} \quad (7)$$

where c_{ij} is the cost per unit distance from i and j; d_{ij}^{x} is the rectilinear distance of the centroids from departments i and j on the x-axis; d_{ij}^{y} is the rectilinear distance of the centroids from i and j on the y-axis; λ is the relative importance of penalty costs and $\lambda=\sum_i\sum_j 10 f_{ij}c_{ij}WH$ [18]; $[\ ]^{+}$ denotes returning a positive value of a subtraction expression or zero, i.e. $[\ ]^{+}=\max\{0,a-b\}$; Lb_i^{h} is the lower height limit of department i; Lb_i^{w} is the lower width limit of i; Ub_i^{h} is the upper height limit of i and Ub_i^{w}; and Ub_i^{w} is the upper width limit of i and Ub_i^{h}.

5. Procedural Steps

Based on the mechanisms of ACS and CSA, the overall procedure for the IACS-FBS is given in this section. It includes standard procedures of ACS, i.e., parts of Step 0 (except for Step 0.2), Step 2 (only N initial ant solutions are needed), Step 3, Step 9, parts of Step 10 (except for Steps 10.1, 10.4, and 10.10), Step 13, and Step 14. Note that ant solutions are constructed using the space filling heuristic having the most proper bay number in Step 2. That modification is made to achieve better initial solutions.

The rest of the IACS algorithm is developed utilizing the CSA. CSA is based on the clonal selection principle. The antibody clone and fitness-related mutation are the two remarkable features of the CSA. First, a temporary pool is generated in Steps 1 to 7. The size of the temporary pool is two times the number of the colony. In Step 4, certain ants are reselected because of their diversity for the current best solution to maintain ant diversity. For the same consideration, mutated ants are generated in Step 5. Two mutation operations are performed: Swap between a department sequence, which exchanges the positions of two departments in the department sequence, and switch of a bay break, which conditionally changes the value of a bay break code from 0 to 1 or 1 to 0. That is, the first and the last bay break code are fixed and the sum of three successive values of bay break codes must be less than or equal to 1.

Next, all solutions in the temporary pool are selected for the ant colony in Step 8. Then the ant colony is further improved by an optimization search in Step 9 and by local searching in Step 10. Step 10 is different from the standard ACS. We do not perform a local search for all ant solutions, local searching is applied to the best solution of the iteration to further improve it. Herein, we regard a local search as a mutation operation. The mutation rate for each ant is inversely proportional to its fitness.

After mutation, the ant colony is put into a mutated ant pool. Applying the mutated ants pool, a memory pool and a candidate pool are updated in Steps 12 and 13, respectively. Note that we do not allow identical ants in either the memory pool or the candidate pool in order to increase ant diversity.

The detailed steps for the IACS-FBS are listed herein. The flowchart is given in Fig. 2.

Step 0: Parameter setting and initialization
Step 0.1: Set algorithm parameters of ACS, maximum number of iterations (*NI*), number of ants (*N*), pheromone information parameter (α), heuristic information parameter (β), and evaporation rate (ρ).
Step 0.2: Set algorithm parameters for CAS, size of memory pool ($r = N \times b\%$), clone number of best ant-solutions ($s_1 = (N - r) \times d\%$), and clone number of diverse ant-solutions ($s_2 = (N - r) \times (1 - d\%)$).
Step 0.3: Initialize iteration number counter. Set $I := 0$.
Step 0.4: Initialize pheromone information $\tau_{ij}^{0}, \forall i, j$.
Step 0.5: Initialize the fitness value for the global best solution. Set $z^* = \infty$.

Step 1: Generate an empty memory pool *M*

Step 2: Generate initial candidate pool *P* of ant colony (2*N* ants)
Step 2.1: Initialize ant number counter. Set $p = 0$.
Step 2.2: Initialize the fitness value of the iteration best solution. Set $z_I^* = \infty$.
Step 2.3: Update ant number counter $p = p + 1$.
Step 2.4: Perform the ant solutions construction procedure [22] to create ant *p*.
Step 2.5: If the number of ants is less than 2*N*, then go to Step 2.3.

Step 3: Evaluate the fitness of the ant colony in the candidate pool *P*

Step 4: Generate a temporary pool *C* from the memory pool *M* and the candidate pool *P*
Step 4.1: Clone the ants in memory pool *M* (*r* ants) into the temporary pool *C*.
Step 4.2: Clone the best ants in candidate pool *P* (s_1 ants) into the temporary pool *C*.
Step 4.3: According to Eq. (8), evaluate the diversity measurement between each ant and the best ant in the candidate pool *P*.

$$\delta = \sum_l \left| b_l - b_l^* \right| \tag{8}$$

where b_l and b_l^* is the current and the best bay width of bay *l*, respectively.
Step 4.4: Clone the diverse ants in the candidate pool *P* (s_2 ants) into the temporary pool *C*.

Step 5: Generate a mutated ants pool *C*1 from the temporary pool *C*
Perform mutation operations for a department sequence and/or a bay break to all ants in the temporary pool *C*.

Step 6: Evaluate the fitness of all ants in the mutated ant pool *C*1

Step 7: Update the temporary pool *C*
If the mutated ant is better than the original ant, then replace the original ant.

Step 8: Select an ant colony (*n* ants) from the temporary pool *C*

Step 9: Perform optimization search of ant colony
Step 9.1: Exploit the selected regions by sending the ants on a local search by performing a state transition rule.
Step 9.2: Update the local pheromone for all the ants.

Step 10: Mutate the current ant solutions for this iteration and perform the local search operations [22] to the current best solution
Step 10.1: Determine the threshold of the mutation rate ϕ by Eq. (9).

$$\phi = N \Big/ \sum_p z_p \tag{9}$$

Step 10.2: Initialize the ant number counter. Set $p = 0$.
Step 10.3: Update the ant number counter $p = p + 1$.
Step 10.4: Calculate the mutation rate of ant *p*, $\phi_p = 1/z_p$. If the value ϕ_p is less than the threshold ϕ, go to Step 10.5; otherwise, go to Step 10.10.

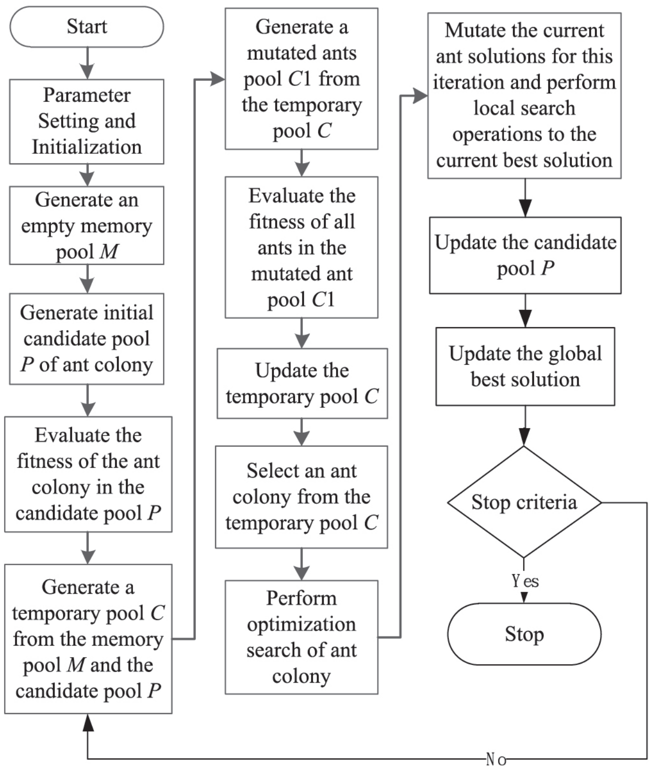

Fig. 2 Flowchart of IACS approach

Step 10.5: Perform the local search operations [22] for a department sequence and a bay break to the ant solution p.
Step 10.6: Calculate the fitness value $\hat{z}_I$ of the ant p after a local search.
Step 10.7: Update the best solution of this iteration z_I^*, once a new best solution is found ($\hat{z}_I < z_I^*$).
Step 10.8: Update the local pheromone of the mutated ant p, if its fitness value is improved.
Step 10.9: Update the global pheromone of the mutated ant p, if its fitness value is not improved.
Step 10.10: Add ant p or mutated ant p to the mutated ants pool $C1$.
Step 10.11: If the number of ants is less than N, then go to Step 10.3; otherwise continue.

Step 11: Update the memory pool M
Step 11.1: Clone the best r ants in the mutated ants pool $C1$ into the memory pool M.
Step 11.2: Delete identical ants in the memory pool M

Step 12: Update the candidate pool P
Step 12.1: Delete identical ants in the candidate pool P to maintain ant diversity.
Step 12.2: Replace those ant solutions in the candidate pool P with rest ants with better fitness in the mutated ants pool $C1$.

Step 13: Update the global best solution
If z_I^* is less than z^*, update the fitness value of the global best solution $z^* := z_I^*$.

Step 14: Stop criteria
If the maximum number of iterations is realized, then output the global best solution and stop; otherwise, go to Step 4.

IV. COMPUTATIONAL EXPERIMENTS

The proposed algorithm was tested using several problem sets, as listed in Table 1. Note that M11 was modified to allow the use of FBS representation. The location of the last department was fixed by assigning it to the last position of the facility, but the department fixed size constraint was relaxed. The algorithm was coded with C^{++} and tested using an Intel Core i7 CPU processor with 4G RAM.

Based on previous research and pre-tuning conducted by this study, all parameter values were determined. We set a maximum number of iterations, an evaporation rate, and a probability for choosing a solution component (q_0), equal to 500, 0.1, and 0.5, respectively. All algorithm parameters for these nine testing examples are offered in Table 2. As the problem size increases, the number of ants and maximum number of iterations will increase.

Table 3 details the previously best-known results for the test problems. The algorithm was replicated 10 times, and the IACS-FBS results are then compared to other FBS solutions. The comparative results show that the IACS-FBS approach is indeed very promising. For the Nug15a5 problem, the IACS-FBS found a new best FBS solution as shown in Fig. 3.

For problem sets with fewer than 10 departments, our solutions are better than or similar to those obtained using

Table 1 Problem set data

Prob. set	No. of Dpt.	Facility size		Maximum aspect ratio
		Width	Height	
O7	7	8.54	13.00	$\alpha^{max} = 4$
O8	8	11.31	13.00	$\alpha^{max} = 4$
FO7	7	8.54	13.00	$\alpha^{max} = 5$
FO8	8	11.31	13.00	$\alpha^{max} = 5$
O9	9	12.00	13.00	$\alpha^{max} = 4, 5$
vC10a	10	25.00	51.00	$\alpha^{max} = 5$
M11a	11	3.00	2.00	$\alpha^{max} = 5$
Nug12	12	3.00	4.00	$\alpha^{max} = 5$
Nug15	15	3.00	5.00	$\alpha^{max} = 5$

Table 2 Parameters setting

Problem set	No. of Ants	Pheromone information parameter	Heuristic information parameter
O7, O8, FO7, FO8, O9, vC10a	100	3	1
M11a	150	5	5
Nug12	200	5	5
Nug15	400	5	5

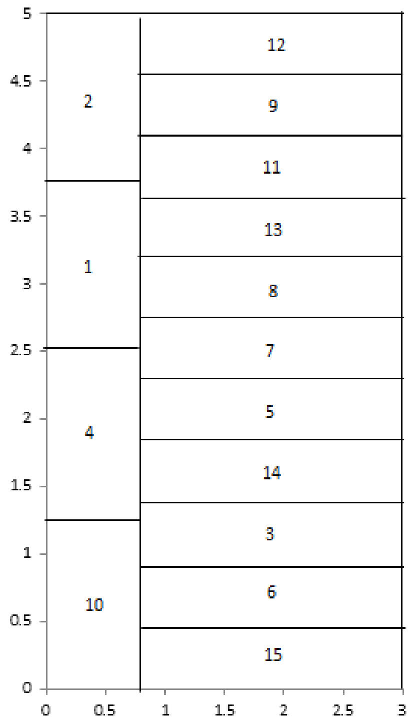

Fig. 3 A new solution for Nug15a5

Table 3 Comparisons of best FBS solutions to other approaches

Problem	Shape Cons.	Konak *et al.* [4]	Wong and Komarudin [17]	Kulturel-Konak and Konak [18]	Kulturel-Konak and Konak [3]	Chang and Lin [22]	This Study
O7a4	α^{max}=4	-	136.58	-	-	134.19*	134.19*
O8a4	α^{max}=4	-	-	-	-	245.51*	245.51*
O9a4	α^{max}=4	-	-	-	-	241.06*	241.06*
O9a5	α^{max}=5	241.06	241.06	-	-	238.12*	238.12*
FO7a5	α^{max}=5	23.12	23.12	-	-	18.88*	18.88*
FO8a5	α^{max}=5	22.39*	22.39*	-	-	22.39*	22.39*
vC10a5	α^{max}=5	21,463.07	21,463.1	21,463.07	20,142.13*	20,142.13*	20,142.13*
M11a5	α^{max}=5	1,225.00	1,204.15	-	1,201.12*	1,201.12*	1204.15
Nug12a5	α^{max}=5	265.6	262*	-	-	262*	262*
Nug15a5	α^{max}=5	526.75	536.75	-	-	-	523.67*
Approach	-	MIP+FBS	AS+FBS	ACO+FBS	PSO+FBS	ACS+FBS	IACS+FBS

MIP or AS. Compared to the results of ACS, the solution quality of IACS is the same as or ACS. The required CPU times or ACS and IACS are shown in Fig. 4. We observed that the algorithm speed of IACS is faster than for ACS. For problem sets with more than 9 departments, our solutions are better than or similar to those solutions obtained by MIP, AS, ACO, or PSO, except for M11a5. For the M11a5 problem, our solution is better than or similar to the solutions obtained by MIP, or AS. Required CPU times for ACS and IACS are shown in Fig. 5. The algorithm speed of IACS is still faster than ACS; however, the gap between them is smaller.

V. DISCUSSIONS AND CONCLUSIONS

To prevent a premature convergence problem and escape from a local optimal solution, cloning with affinity-related mutation of the CSA is utilized and combined with the ACS in this algorithm. In this study, an IACS-FBS algorithm is proposed to solve the unequal area FLP. We regard local searching as a mutation operation and the mutation rate is inversely proportional to its fitness. In addition, the diversities between the current best solution are measured to help choose clone candidates. Identical ants in a memory pool and a candidate pool are deleted to maintain diverseness among the ant colony. For problem Nug15a5, a new best FBS solution is found.

CSA mimics parallel hill climbing, so it can exploit and explore the solution space in parallel terms and effectively [29]. Parallel search methods perform better than do strictly serial searches and randomized search methods. IACS-FBS can provide more efficient and more comprehensive exploitation and exploration, including the construction of initial solutions and local search methods. Introducing CSA operations in the ant colony system can improve the convergence speed of the ant colony system and increase the differences among ant solutions. The search capability of the immune ant colony system is thus enhanced. Compared to existing ACS algorithms, the proposed algorithm obtains better or at least the same solution quality, but in a shorter time, except for problem M11a5. General speaking, the algorithm speed of IACS is faster than ACS.

Moreover, the heuristic information function should be modified to improve the efficiency of the IACS. Finally,

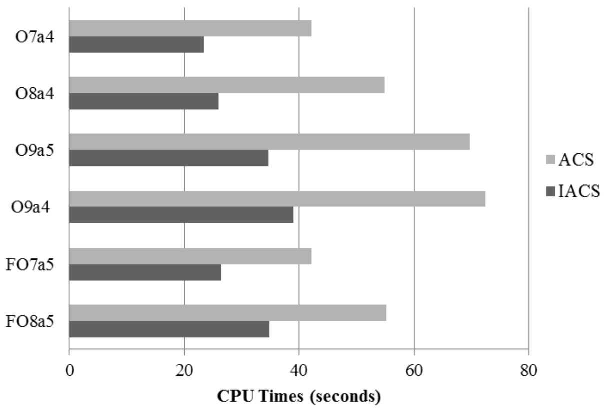

Fig. 4 Comparisons of ACS and IACS in computing times (I)

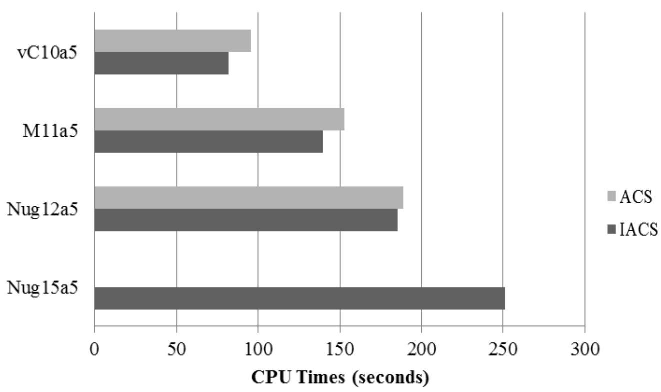

Fig. 5 Comparisons of ACS and IACS in computing times (II)

this study can be expanded in the future effectively to consider other restrictions, such as fixed location departments and fixed size departments. This future work should also include identifying more complicated local searches in order to have better results for the medium and large instances.

ACKNOWLEDGEMENTS

This work was partly supported in part by the National Science Council, Taiwan, under Grants NSC 99-2410-H-033-025-MY2.

REFERENCES

[1] X. Tong, "SECOT: A sequential construction technique for facility design," Ph.D. thesis, University of Pittsburgh, USA, 1991.
[2] O. Alagoz, B. A. Norman, and A. E. Smith, "Determining aisle structures for facility designs," *IIE Transactions*, vol. 40, no. 11, pp. 1019-1031, 2008.
[3] S. Kulturel-Konak and A. Konak, "A new relaxed flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem," *Engineering Optimization*, vol. 43, pp. 1-25, 2011b.
[4] A. Konak, S. Kulturel-Konak, B. A. Norman, and A. E. Smith, "A new mixed integer formulation for optimal facility layout design," *Operation Research Letters*, vol. 34, pp. 660-672, 2006.
[5] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: WH Freeman, 1979.
[6] I. Castillo, J. Westerlund, S. Emet, and T. Westerlund, "Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization methods," *Computers and Chemical Engineering*, vol. 30, pp. 54-69, 2005.
[7] A. R. Mckendall Jr. and J. Shang, "Hybrid ant systems for the dynamic facility layout problem," *Computers and Operations Research*, vol. 33, no. 3, pp. 790-803, 2006.
[8] P. Banerjee, Y. Zhou, and B. Montreuil, "Genetically assisted optimization of cell layout and material flow path skeleton," *IIE Transactions*, vol. 29, no. 4, pp. 277-291, 1997.
[9] K. Y. Gau and R. D. Meller, "An iterative facility layout algorithm," *International Journal of Production Research*, vol. 37, no. 16, pp. 3739-3758, 1999.
[10] B. A. Norman, R. A. Arapoglu, and A. E. Smith, "Integrated facilities design using a contour distance metric," *IIE Transactions*, vol. 33, no. 4, pp. 337-344, 2001.
[11] D. M. Tate and A. E. Smith, "Unequal-area facility layout by genetic search," *IIE Transactions*, vol. 27, pp. 465-472, 1995.
[12] S. Kulturel-Konak, B. A. Norman, D. W. Coit, and A. E. Smith, "Exploring tabu search memory in constrained problems," *INFORMS Journal on Computing*, vol. 16, pp. 241-254, 2004.
[13] S. Kulturel-Konak, A. E. Smith, and B. A, Norman, "Bi-objective facility expansion and relayout considering monuments," *IIE Transactions*, vol. 39, no. 7, pp. 747-761, 2007.
[14] R. Logendran and T. Kriausakul, "A methodology for solving the unequal area facility layout problem using distance and shape-based measures," *International Journal of Production Research*, vol. 44, no. 7, pp. 1243-1272, 2007.
[15] D. Scholz, A. Petrick, and W. Domschke, "STaTS: a slicing tree and tabu search based heuristic for the unequal area facility layout problem," *European Journal of Operational Research*, vol. 197, no. 1, pp. 166-178, 2009.
[16] Komarudin and K. Y. Wong, "Applying ant system for solving unequal area facility layout problems," *European Journal of Operational Research*, vol. 202, no. 3, pp. 730-746, 2010.
[17] K. Y. Wong and Komarudin, "Solving facility layout problems using flexible bay structure representation and ant system algorithm," *Expert Systems with Applications*, vol. 37, pp. 5523-5527, 2010.
[18] S. Kulturel-Konak and A. Konak, "Unequal area flexible bay facility layout using ant colony optimization," *International Journal of Production Research*, vol. 49, no. 7, pp. 1877-1902, 2011a.
[19] B. Haktanirlar Ulutas and S. Kulturel-Konak, "An artificial immune system based algorithm to solve unequal area facility layout problem," *Expert Systems with Applications*, vol. 39, pp. 5384-5395, 2012.
[20] O. Engin and A. Doyen, "Artificial immune systems and applications in industrial problems," *Gazi University Journal of Science*, vol. 17, no. 1, pp. 71-84, 2004.
[21] J. Timmis, P. Andrews, and E. Hart, "On artificial immune systems and swarm intelligence," *Swarm Intelligence*, vol. 4, no. 4, pp. 247-273, 2010.
[22] M. S. Chang and H. Y. Lin, "A flexible bay structure representation and ant colony system for unequal area facility layout problems," *Lecture Notes in Engineering and Computer Science*, vol. 2199, no. 1, pp. 1346-1351, 2012.
[23] L. M. Gambardella, E. D. Taillard, and M. Dorigo, "Ant colonies for the quadratic assignment problem," IDISIA, Dept. of Comput. Sci. Tech. Rep., pp. 4-97, 1997.
[24] E. G. Talbi, O. Roux, C. Fonlupt, and D. Robillard, "Parallel ant colonies for the quadratic assignment problem," *Future Generation Computer Systems*, vol. 17, pp. 441-449, 2001.
[25] H. D. Pour and M. Nosraty, "Solving the facility and layout and location problem by ant colony optimization-metaheuristic," *International Journal of Production Research*, vol. 44, no. 23, pp. 5187-5196, 2006.
[26] Y. Hani, L. Amodeo, F. Yalaoui, and H. Chen, "Ant colony optimization for solving an industrial layout problem," *European Journal of Operational Research*, vol. 183, no. 2, pp. 633-642, 2007.
[27] M. Nourelfath, N. Nahas, and B. Montreuil, "Coupling ant colony optimization and the extended great deluge algorithm for the discrete facility layout problem," *Engineering Optimization*, vol. 39, no. 8, pp. 953-968, 2007.
[28] Komarudin, "An improved ant system algorithm unequal area facility layout problems," Master thesis, University of Teknologi, Malaysia, 2009.
[29] S. Gao, W. Wang, H. Dai, F. Li, and Z. Tang, "Improved clonal selection algorithm with ant colony optimization," *IEICE Transactions Information and Systems*, vol. 6, pp. 1813-1823, 2008.