

An Immunized Ant Colony System Algorithm to Solve Unequal Area Facility Layout Problems Using Flexible Bay Structure

Mei-Shiang Chang and Hsin-Yi Lin

Abstract The Facility Layout Problem (FLP) is a typical combinatorial optimization problem. In this research, clonal selection algorithm (CSA) and ant colony system (ACS) are combined and an immunized ant colony system algorithm (IACS) is proposed to solve unequal-area facility layout problems using a flexible bay structure (FBS) representation. Four operations of CSA, clone, mutation, memory cells, and suppressor cells, are introduced in the ACS to improve the solution quality of initial ant solutions and to increase differences among ant solutions, so search capability of the IACO is enhanced. Datasets of well-known benchmark problems are used to evaluate the effectiveness of this approach. Compared with preview researches, the IACS can obtain the close or better solutions for some benchmark problems.

Keywords Unequal-area facility layout • Ant colony optimization • Clonal selection algorithm • Flexible bay structure • Constrained combinatorial optimization

1 Introduction

Facility layout problems (FLPs) aim to find the optimal arrangement of a given number of non-overlapping departments with unequal area requirements within a facility and certain ratio constraints or minimum length constraints. The common objective is to minimize the total material handling costs among departments.

M.-S. Chang (✉) · H.-Y. Lin

Department of Civil Engineering, Chung Yuan Christian University, 200, Chung Pei Road,
Chung Li 32023, Taiwan, Republic of China
e-mail: mschang@cycu.edu.tw

H.-Y. Lin

e-mail: ac-1722@hotmail.com

Recently, different ACO approaches have been used to solve various versions of FLP problems. Most of them formulate FLP as a quadratic assignment problem (QAP) and obtain promising solutions to several test problems (Baykasoglu et al. 2006; Mckendall and Shang 2006; Nourelfath et al. 2007; Hani et al. 2007). Such approaches need modification in solving FLP. In addition, an ant system approach was first presented to solve the FLP (Wong and Komarudin 2010; Komarudin and Wong 2010). These algorithms use a FBS and a slicing tree structure to represent the FLP respectively. The former also presents an improvement to the FBS representation by using free or empty space. The algorithm can improve the best known solution for several problem instances. The latter one integrates nine types of local search to improve the algorithm performance. No doubt this heuristic shows encouraging results in solving FLP. Moreover, an ACO is proposed to solve the FLP with FBS (Kulturel-Konak and Konak 2011a, b). Compared with meta-heuristics such as GA, TS, AS, and exact methods, this ACO approach is shown to be very effective in finding previously known best solutions and making notable improvements. Then an ACS is used to solve the FLP with FBS (Chang and Lin 2012). Compared with the previously best known solutions, the ACS can obtain the same or better solutions to some benchmark problems. Such interesting results inspire us to further explore the capability of applying ACS to solve the FLP.

Generally speaking, fusion of the computational intelligence methodologies can usually provide higher performances over employing them separately. This study proposes an immunized ant colony system (IACS) approach to solve the FLP with the flexible bay structure (FBS). It is based on clonal selection algorithm (CSA) and ACS.

2 Immunized Ant Colony System Algorithm

2.1 Solution Representation

We adopt the ant solution representation proposed by Komarudin (2009) for solving FLPs. Each ant solution consists of two parts: the department sequence codes and the bay break codes, such as (1-4-5-7-2-3-6)–(0-0-1-0-0-1). The former represents the order of n departments, which 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 the departments are placed from left to right and bottom to top.

Komarudin (2009) presented this intuitive rule: “A department with higher material flow should be located nearer to the center of the facility.” The heuristic information function was defined by Eq. (3).

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

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.

2.2 Procedure Steps

Based on the mechanisms of ACS and CSA, we propose a hybrid optimization algorithm and name it immunized ant colony system (IACS) algorithm. The overall procedure of the IACS-FBS is given below. It includes standard procedures of ACS, i.e. parts of Step 0 (except Step 0.2), Step 2 (only N initial ant solutions are needed), Step 3, Step 9, parts of Step 10 (except Steps 10.1, 10.4, and 10.10), Step 13, and Step 14. Basically, we extend the study of ACS-FBS proposed by Chang and Lin (2012) except for several minor modifications. In Step 2, ant solutions are constructed by the space filling heuristic with having the most proper bay number. Such a modification is made for achieving better initial solutions.

The rest of the IACS algorithm is developed according to clone selection algorithm. First, a temporary pool is generated in Steps 1–7. The size of the temporary pool is two times the number of the ant colony. In Step 4, certain ants are reselected because of its diversity with the current best solution in order to maintain the 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 conditional changes the value of a bay break code from 0 to 1 or 1 to 0. The first and the last bay break codes are fixed. 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 optimization searching in Step 9 and by local searching in Step 10. It is different to the standard ACS in this step. We don't perform a local search to all ant solutions. We regard a local search as a mutation operation. The mutation rate of each ant is inversely proportional to its fitness. After the ant colony is put back a mutated ant pool. According to the mutated ants pool, a memory pool and a candidate pool are updated in Steps 12 and 13 respectively. Note that we don't allow identical ants in the memory pool and the candidate pool in order to increase the ant diversity.

The detailed steps of the IACS-FBS are listed herein.

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 of CAS, size of memory pool ($r = N \times b \%$), clone number of the best ant-solutions ($s_1 = (N - r) \times d \%$), and clone number of the 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 of 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 ($2N$ ants) by performing the modified space filling heuristic proposed by Chang and Lin (2012)

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 a procedure of ant solutions construction to create ant p .

Step 2.5: If the number of ants is less than $2N$, then go to Step 2.3; otherwise continue.

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

$$z = \sum_i \sum_j f_{ij} c_{ij} (d_{ij}^x + d_{ij}^y) + \lambda \sum_i [Ub_i^w - w_i]^+ + [Lb_i^w - w_i]^+ + \lambda \sum_i [Ub_i^h - h_i]^+ + [Lb_i^h - h_i]^+ \quad (2)$$

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 $\lambda = \sum_i \sum_j 10 f_{ij} c_{ij} WH$; $[\cdot]^+$ denotes returning a positive value of a subtraction expression or zero, i.e. $[\cdot]^+ = \max\{0, a - b\}$; 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^h is the upper width limit of i and Ub_i^h .

Step 4: Generate a temporary pool C from the memory pool M and the candidate pool P

Step 4.1: Clone 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. (3), evaluate the diversity measurement of each ant between the best ant in the candidate pool P .

$$\delta = \sum_l |b_l - b_l^*| \quad (3)$$

where b_l is the current bay widths; b_l^* is the best bay widths.

Step 4.4: Clone the diverse ants in candidate pool P (s_2 ants) into the temporary pool C .

Step 5: Generate a mutated ants pool $C1$ from the temporary pool C

Perform mutation operations of a department sequence and/or of a bay break to all ants in the temporary pool C .

Step 6: According to Eq. (1), evaluate the fitness of all ants in the mutated ant pool $C1$

Step 7: Update the temporary pool C

If the mutated ant is better than the original one, the original ant is replaced.

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

Step 9: Optimization searching of ant colony

Step 9.1: Exploit the selected regions by sending the ants for local search by performing a state transition rule.

$$j = \begin{cases} \arg \max [\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta, & \text{if } q \leq q_0 \text{ (exploitation)} \\ S, & \text{otherwise (exploration)} \end{cases} \quad (4)$$

$$S = P_{ij}^k = \begin{cases} [\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta / \sum_{q \in N_i} [\tau_{iq}]^\alpha \cdot [\eta_{iq}]^\beta, & \text{if } j \in N_i \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

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 ($0 \leq q_0 \leq 1$); α is a parameter which determines the relative weight of pheromone information; and β is a parameter which 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 available alternatives of the department sequence to be chosen by the corresponding ant located in department i .

Step 9.2: Update the local pheromone all the ants according to Eq. (6).

$$\tau_{ij} := (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \quad (6)$$

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

Step 10: Mutate the current ant solutions of this iteration by performing the proposed local searching. The mutation rate of each ant is inversely proportional to its fitness, that is, the mutation rate is proportional to the affinity

Step 10.1: Determine the threshold of mutation rate ϕ by Eq. (7).

$$\phi = N / \sum_p z_p \quad (7)$$

Step 10.2: Initialize ant number counter. Set $p = 0$.

Step 10.3: Update 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.

Step 10.5: Perform local search operations of a department sequence (swap, one-insert, or two-opt, Chang and Lin 2012) and of a bay break to the ant solution p .

Step 10.6: Calculate the fitness value \hat{z}_I of the ant p after 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 according to Eq. (6), if its fitness value is improved.

Step 10.9: Update the global pheromone of the mutated ant p according to Eqs. (8) and (9), if its fitness value is not improved.

$$\tau_{ij} := (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{p \in P} \quad (8)$$

$$\Delta\tau_{p \in P} = \sum_{ij \in P} d_{ij}^x + d_{ij}^y \quad (9)$$

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 the ant diversity.

Step 12.2: Replace those ant solutions in the candidate pool P by the 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: Stopping criteria

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

3 Computational Experiments

The proposed algorithm was tested using several problem sets, as listed in Table 1. Note that M11a and M15a were modified to allow the use of FBS representation. The location of the last department is fixed by assigning it to the last position of the facility, but the department fixed size constraint is relaxed. The algorithm was coded with C++ and tested using an Intel(R) Core(TM) i7 CPU processor.

Table 2 provides the previous best-known results of the test problems. The IACS-FBS results are compared to other FBS solutions. The comparative results show that the ACS-FBS approach is very promising. For problem Nug15a5, the IACS-FBS found a new best FBS solution.

4 Conclusions

To prevent the premature convergence problem and to escape from a local optimal solution, clone 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 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 the diverseness among the ant colony. Furthermore, we revise the ACS-FBS to provide more efficient and comprehensive local exploitation, such as the construction of initial solutions and the local search methods. Compared with existing ACO algorithms, the proposed algorithm obtain better or at least the same solution quality, except for problem Nug15a4. For problem Nug15a5, a new best FBS solution is found.

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	$\alpha^{\max} = 4, 5$
M15a	15	15.00	15.00	$\alpha^{\max} = 5$
Nug12	12	3.00	4.00	$\alpha^{\max} = 4, 5$
Nug15	15	3.00	4.00	$\alpha^{\max} = 4, 5$

Table 2 Comparisons of the best FBS solutions with other approaches

Problem	Shape cons.	Konak et al. (2006)	Wong and Komarudin (2010)	Kulturel-Konak and Konak (2011a)	Kulturel-Konak and Konak (2011b)	Chang and Lin (2012)	This study
O7	$\alpha^{max} = 4$	—	136.58	—	—	134.19*	134.19*
O8	$\alpha^{max} = 4$	—	—	—	—	245.51*	245.51*
O9a4	$\alpha^{max} = 4$	—	—	—	—	241.06*	241.06*
O9a5	$\alpha^{max} = 5$	241.06	241.06	—	—	238.12*	238.12*
FO7	$\alpha^{max} = 5$	23.12	23.12	—	—	18.88*	18.88*
FO8	$\alpha^{max} = 5$	22.39*	22.39*	—	—	22.39*	22.39*
vC10a	$\alpha^{max} = 5$	21,463.07	21,463.1	21,463.07	20,142.13*	20,142.13*	20,142.13*
M11a4	$\alpha^{max} = 4$	—	—	1,268.55	—	—	1268.55*
M11a5	$\alpha^{max} = 5$	1,225.00	1,204.15	—	1,201.12*	1,201.12*	1271.56 SingYi
Nug12a4	$\alpha^{max} = 4$	265.5	—	262	257.5*	262	1204.15 black
Nug12a5	$\alpha^{max} = 5$	265.6	262*	—	—	262*	262*
M15a	$\alpha^{max} = 5$	31,779.09	27,545.30	—	27,545.27	—	27701.01*hm
Nug15a4	$\alpha^{max} = 4$	526.72	—	524.75	524.75	—	542.8i7
Nug15a5	$\alpha^{max} = 5$	526.75	536.75	—	—	—	532.77sil
Approach	—	MIP + FBS	AS + FBS	ACO + FBS	PSO + FBS	ACS + FBS	IACS + FBS

References

- Baykasoglu A, Dereli T, Sabuncu I (2006) An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. *Omega* 34(4):385–396
- Chang MS, Lin HY (2012) A flexible bay structure representation and ant colony system for unequal area facility layout problems. *Lecture Notes Eng Comp Sci* 2199(1):1346–1351
- Hani Y, Amodeo L, Yalaoui F, Chen H (2007) Ant colony optimization for solving an industrial layout problem. *Eur J Oper Res* 183(2):633–642
- Komarudin (2009) An improved ant system algorithm unequal area facility layout problems. Master Thesis, University of Teknologi, Malaysia
- Komarudin, Wong KY (2010) Applying ant system for solving unequal area facility layout problems. *Eur J Oper Res* 202(3):730–746
- Konak A, Kulturel-Konak S, Norman BA, Smith AE (2006) A new mixed integer formulation for optimal facility layout design. *Oper Res Lett* 34:660–672
- Kulturel-Konak S, Konak A (2011a) Unequal area flexible bay facility layout using ant colony optimization. *Inter J Prod Res* 49(7):1877–1902
- Kulturel-Konak S, Konak A (2011b) A new relaxed flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem. *Eng Optimiz* 43:1–25
- Mckendall AR Jr, Shang J (2006) Hybrid ant systems for the dynamic facility layout problem. *Comp Oper Res* 33(3):790–803
- Nourelfath M, Nahas N, Montreuil B (2007) Coupling ant colony optimization and the extended great deluge algorithm for the discrete facility layout problem. *Eng Optimiz* 39(8):953–968
- Wong KY, Komarudin (2010) Solving facility layout problems using flexible bay structure representation and ant system algorithm. *Exp Syst Appl* 37:5523–5527

Proceedings of the Institute of Industrial Engineers
Asian Conference 2013

Lin, Y.-K.; Tsao, Y.-C.; Lin, S.W. (Eds.)

2013, XXIV, 1593 p. 434 illus., 262 illus. in color. In 2
volumes, not available separately., Hardcover

ISBN: 978-981-4451-97-0