

Hybrid optimisation method for the facility layout problem

Yasmina Hani, Lionel Amodeo, Farouk Yalaoui and Haoxun Chen
University of Technology of Troyes (ICD FRE CNRS 2848)
France

1. Introduction

Due to increased competition, the manufacturers are brought to quickly respond to the needs of customers for on-time delivery of high quality products. Many studies have been made to realize these objectives such as he studies on the facility layout problem (FLP).

It is to find a good configuration of the machines, equipments or other resources in a given facility in order to optimize the production flows while minimizing the total cost. The FLP arises in many industry cases, such as facility reorganization, construction of new production units, or equipment assignment.

To solve this problem, we develop an ant colony optimization algorithm combined with a local search method. This chapter presents the algorithm, its performance evaluation et application to an industrial case.

The remainder of this chapter is organised as follows: In section 2, the layout problem is described. In section 3, we present the general framework of the ACO algorithm and its enhancement by guided local search. Computational results on the performance evaluation of the algorithm using a set of benchmark instances are presented in section 4, and the application of the algorithm to an industrial case is reported in section 5. In section 6, we conclude this chapter with some remarks.

2. The layout problem

The layout problem is commonly met in the industry. A full description of the problem can be found in (Kusiak & Heragu, 1987). Layout problem is known to be NP-Hard (Sahni & Gonzales, 1976).

There are many cases where FLP is considered in facility reorganization, construction of new production units, or equipment assignment. Layout problem could be also found in many classical and theoretical studies. However, only few layout industrial cases are treated in the literature. Hicks (2004), developed a genetic algorithm for minimizing material movement in a manufacturing cell with application to practical problems related to the capital good industry. Lee *et al.* (2002) proposed a genetic algorithm for solving multi-floor facility layout problems with the facility's inner structure consisting of walls and passages. A study related to the fashion industry was presented by Martens (2004).

Source: Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Book edited by: Felix T. S. Chan and Manoj Kumar Tiwari, ISBN 978-3-902613-09-7, pp. 532, December 2007, Itech Education and Publishing, Vienna, Austria

The FLP has been formulated as a Quadratic Assignment Problem (QAP). Other formulations also exist such as mixed integer programming (Meller et al., 1999) and graph theoretic model (Caccetta & Kusumah, 2001).

Many methods are used to solve the layout problem and they are essentially based on meta-heuristics such as Genetic Algorithms (GA) (Lee et al., 2002), Tabu Search (Chiang & Chiang, 1998), Simulated Annealing (SA) (Baykasoglu & Gindy, 2001), Ant Colony (Solimanpur et al., 2004). In our study, we consider a layout problem modelled as a QAP.

3. Hybrid Ant Colony Optimization

3.1 Ant colony optimization (ACO)

The principle of ACO algorithms (Corne & Dorigo, 1999; Dorigo et al., 2000) is based on the way ants search for food. Each ant takes into consideration (probabilistic choice) pheromone trails left by all other ant colony members which preceded its course, the pheromone trail being a trace, a smell left by every ant on its way. This pheromone evaporates with time, and therefore the probabilistic choice for each ant changes with time. After many ant courses, the path to the food will be characterized by higher pheromone traces and thus all ants will follow the same path. This collective behaviour, based upon a shared memory among all colony ants could be adapted and used for solving combinatorial optimization problems with the following analogies:

- The real ant search space becomes the space of the combinatorial problem solutions.
- The amount of food inside a source becomes the evaluation of the objective function for the corresponding solution.
- The pheromone trails become an adaptive shared memory.

Ant colony optimization (ACO) problems could therefore be encoded as finding the shortest path in a graph. One of the first applications of ACO was the travelling salesman problem.

In the general case, the ant colony algorithm applies the artificial ants' concept; it is represented by the following steps:

Step 1: Initialization of parameters

Step 2: Construction of solutions

Step 3: Local search algorithm

Step 4: Pheromone updating rule

Step 5: Return to 2 until a given stopping criterion satisfied

Ant colony optimization (ACO) is a method widely used for solving quadratic assignment problem. The first application was proposed by Maniezzo et al. in 1994. Since that, many applications were proposed, and the differences were in the generation of solutions, the local search method and the pheromone updating. Stutzle & Dorigo (1997) reviewed the ant algorithm applied to solve QAP and reported that the ACO algorithms are among the well performing methods to solve QAP. The MAX-MIN ant system algorithm (MMAS) proposed by Stutzle & Hoos (2000) allows only the best solution to add pheromone trail during the pheromone trail update. A bound is used for trail levels to avoid premature convergence of the search. Gambardella et al (1997) proposed a hybrid ant system HAS-QAP to solve QAP. The originality of their approach is in that the pheromone trail was not used to construct solutions but to modify them in the local search.

Most of the proposed metaheuristics for the FLP problem are effective for small instances. Their performances become worse with the increase of the problem size (i.e. number of resources). Solimanpur et al. (2004) proposed an ACO algorithm for the inter-cell layout

problem formulated as QAP. They proposed a technique based on the partial contribution of each assignment for the calculation of a lower bound used in (Maniezzo, 1999). It was limited to only 30 departments because of the complexity of the problem. In a previous study, ANTabu (Stützle & Dorigo, 1999) using an ant colony optimization with a tabu local search procedure, was successfully applied to the QAP for large instances (i.e. 256 resources).

The ant colony algorithm adapted to the layout problem is composed of the following elements:

1. Construction of solutions

In the proposed algorithm, it is assumed that each ant initially assigns a task i to location j noted (i,j) , then another task to another location k , and so on until a complete solution is obtained. A tabu list represents the set of tasks that the ants has already assigned, the list of the couples (i,j) . This list ensures that all the tasks are assigned to locations. The criterion of the tasks assignment takes into account the probability of assignment with a given site, and depends on two terms, one relating to each ant (visibility) and the other relating to the quantity of pheromones deposited by the whole of the ants.

2. Heuristic information

The ants are not completely blind, they calculate the cost relating to the assignment of a task to a given site. This cost takes into account the flow and distances matrix. Heuristic information, called *visibility* is a function of the assignment cost. Several formulas were used in the literature and each one is adapted to a given problem. Concerning QAP, the assignment of a task i to the site l depends on the tasks assigned before. We define the cost associated with the assignment (i,l) as:

$$C(i,l) = \sum_{s=1}^{l-1} (f_{r(s)i} \times d_{sl} + f_{ir(s)} \times d_{ls}) \quad (1)$$

where, r denotes a permutation of resources under construction. The visibility which represents the desirability of move is defined as:

$$\eta_{il} = \frac{1}{1 + \sum_{s=1}^{l-1} (f_{r(s)i} \times d_{sl} + f_{ir(s)} \times d_{ls})} \quad (2)$$

The reason for which number 1 is added to the denominator of the fraction in (2) is for avoiding division by 0. This formula means that the assignments with smaller contribution to the objective function would be more desirable for selection.

3. Pheromone updating

The pheromone updating mechanism is represented by the following equation

$$\tau_{il}(t) = \lambda \tau_{il}(t-1) + \sum_k \Delta \tau_{il}^k \quad (3)$$

where, $\tau_{il}(t)$ is the quantity of pheromone associated with the assignment of the task i to location l for each ant k for the iteration t . As an ant chooses this assignment, the quantity $\tau_{il}(t)$ increases. The parameter λ is a scaling factor. A large λ results in quick convergence to a local optima solution.

Finally,

$$\Delta\tau_{il}^k = \sum_k \frac{Bestfit}{fit[k]} \quad (4)$$

denotes the magnitude of change in the trail level of an assignment through ant. As seen, the smaller is the fitness solution $fit[k]$ obtained by ant k , the more would be the increment in trail levels selected by ant k .

4. Selection probability

An ant k chooses task i to assign to location l by the following probability:

$$p_{il}^k = \frac{\alpha \times \tau_{il} + (1 - \alpha) \times \eta_{il}}{\sum_{i \neq Tabu_k} (\alpha \times \tau_{il} + (1 - \alpha) \times \eta_{il})}, \quad (5)$$

where, α contributes to make a balance between the choice adopted by the whole of the ants (α near to 1) and the choice of each ant based on its own visibility (α near to 0). We note that a task is assigned to a location if the relative quantity of pheromones is significant or if the associated cost is weak. Finally, the task having the largest probability is assigned to location l .

5. Local search

We choose local search method 2-opt which is simple and well adapted to QAP (Solimanpur et al., 2004). This method applies to a given solution all possible permutations of pairs of tasks. The permutation giving the minimal cost is selected as a local minimum next to the starting solution. This process is repeated until no improvement is observed.

In order to limit computation time during the exchanges, we made the following simplification; if the exchange is done between the elements π_i and π_j of the permutation π , the difference in the objective function value will be then:

$$\begin{aligned} \Delta(\pi, i, j) = & (d_{ii} - d_{jj})(f_{\pi, \pi_j} - f_{\pi, \pi_i}) + (d_{ij} - d_{ji})(f_{\pi, \pi_i} - f_{\pi, \pi_j}) + \\ & \sum_{k \neq i, j} (d_{ki} - d_{kj})(f_{\pi, \pi_j} - f_{\pi, \pi_i}) + (d_{ik} - d_{jk})(f_{\pi, \pi_k} - f_{\pi, \pi_k}) \end{aligned} \quad (6)$$

This algebraic simplification was used by Gambardella et al. in 1997 when they propose HAS-QAP, a hybrid ant colony system applied to the quadratic assignment problem.

The local search does not necessarily lead to a global minimum. In most cases, it converges to a local minimum. For this, a guided local search (GLS) method is used to "penalize" the local minimum found in order to converge to the global minimum. GLS will be explained in detail later.

6. Diversification

Used by Gambardella in 1997, the diversification mechanism is activated if during a number of iterations max_iter , no improvement to the best generated solution is detected. Diversification consists of erasing all the information contained in the pheromone trail by a re-initialization of the pheromone trail matrix and of generating randomly a new current solution for all the ants but one that receives the best solution produced by the search so far. Another possibility is to erase all pheromone trails except for the best solution.

Ant colony algorithm

We propose the following general ant colony optimization algorithm with 2-opt.

Step 1: initialization of parameters for all the tasks and locations

Step 2: for each ant

Assign tasks to locations with a probability p

Update the pheromones

If the best solution is not improved until \max_iter iterations, $\tau_{ij} = 0$, except for the best solution.

Step 3: Return to Step2 until stopping criterion is satisfied.

3.2 Enhancement of ACO by Guided Local Search (GLS)

Guided Local Search (GLS) (Mills and Tsang, 2002) is a metaheuristic which sits as a good local search algorithm. When the given local search algorithm settles in a local optimum, GLS changes the objective function, by increasing penalties in an augmented objective function, associated with *features* contained in the local optimum. The local search then continues to search using the augmented objective function.

The choice of solution features depends on the problem type, and each feature f_i defined must have the following components:

1. An indicator function $I_i(s)$ indicating whether the feature is present in the current solution or not. It is equal to 1 if the feature f_i is present in the solution s and 0 otherwise.
2. A cost function $c_i(s)$ which gives the cost of having f_i in s .
3. A penalty p_i initially set to 0, used to penalize the occurrence of f_i in local minima.

When the local search returns a local minimum s , GLS increases the penalty of the features of s which have maximum utility $util(s, f_i)$ defined as follow :

$$util(s, f_i) = I_i(s) \frac{c_i(s)}{1 + p_i} \quad (7)$$

The idea is to penalise the features, which have highest costs first. GLS uses an augmented cost function (8) in order to guide the local search out of a local optimum. The idea is to make the local minimum more costly than the solutions in the surrounding search space, where the same features are not present.

$$h(s) = g(s) + \lambda' \sum_{i=1}^n I_i(s) \cdot p_i \quad (8)$$

where, $g(s)$ is the cost function and λ' a parameter used to alter the diversification of the search for solutions. A higher value for λ' will result in more diverse search.

The application of GLS for the QAP problem is realised with the following analogies:

- The feature f_{i,π_i} of a solution s corresponds to the assignment of task i to the location π_i .
- The cost related to feature f_{i,π_i} depends on the interaction of the task i with all other tasks of the solution s . This cost is given by

$$C(i, \pi_i) = \sum_{j=1}^n f_{ij} D_{\pi_i, \pi_j} \quad (9)$$

- The value λ' well adapted to the QAP is given by

$$\lambda' = \frac{\sum_{i=1}^n \sum_{j=1}^n f_{ij} \times \sum_{i=1}^n \sum_{j=1}^n D_{ij}}{n^4} \quad (10)$$

The application of GLS technique to the QAP problem could be summarized in the following: Starting from the current solution, a local search method (2-opt for example) is used to find a local minimum, with respect to the augmented cost function. If this minimum has a cost (not augmented) lower than the lowest cost ever found, it is saved as the best ever found solution. Finally, the assignment having the maximum utility would have its corresponding penalty increased.

The GLSQAP algorithm could be summarized as follows:

Step 1: Calculation of λ

Step 2: The best solution $s' = \text{initial solution } s$

Step 3: Perform a local search 2-opt with respect to the augmented cost function, s^ is found as the solution having the lower augmented cost.*

If $\text{cost}(s^) < \text{cost}(s')$, replace s' by s^* .*

Find the assignment (feature) of s^ having the maximum utility, let it be f_{i,π_i} for example.*

Increase the corresponding penalty: $p_{i,\pi_i} = p_{i,\pi_i} + 1$.

Step 4: Return to step 3 until a given stopping criterion is satisfied.

Step 5: s' is the best solution found for the original problem.

Finally, the algorithm procedure of ant colony optimization with GLS is given as follows:

Step 1: initialization of parameters

Step 2: for all ants

a. Assign tasks to locations with the given assignment probability

b. Perform the guided local search GLSQAP

c. Update the pheromones

d. If the best solution is not improved until max_iter iterations, $\tau_{ij} = 0$, except for the best solution.

Step 3: Return to step2 until a stopping criterion is satisfied.

4. Computational results

The algorithm was implemented using Visual C++ 6.0. on a Pentium 3 with 1.8 Ghz CPU speed. In the proposed algorithm four parameters: *ant number AN*, *alpha*, *max_iter* and τ_0 affect the performance of the algorithm. To find the appropriate parameters for our problem, pilot runs were performed. Ant number *AN* was tested between 5 and 60, and a compromise between the quality of the results and the convergence time was found for *AN* = 20. When *AN* was fixed, the best convergence was found for *max_iter* = 10 and *alpha* = 0.6.

Usually, *alpha* is close to 0.5. In our case the value 0.6 indicates that the construction of the solutions more supports the pheromone trails than the individual ant investigation. This value was found to be well adapted with the GLS procedure. Table 1 lists the appropriate values:

Parameter	Value
AN	20
Alpha	0.6
max_iter	10
τ_0	0

Table 1. Parameter values

The performance of this algorithm was tested on instances from the library QAPLIB (Burkard et al., 1991). We first compare our algorithm with the HAS-QAP (Gambardella et al., 1997) method based on ant colonies. We then compare it with ANTabu (Talbi et al., 2001) which is compared with other methods based on genetic algorithms, simulated annealing, tabu search or ant colony and with a recent ant colony optimization algorithm proposed by Solimanpur et al. (2004), which is adapted for problems with a small number of locations. Table 2 compares the results of all the cited algorithms for small instances with a number of locations falling between 19 and 30.

The instances we chose include the regular and irregular problems of QAPLIB. The difference relative to the QAPLIB best known solution is given as a percentage gap. It is almost impossible to have the same experimental settings as for previous studies, but in order to give an idea on the computation time, the mean execution time over 10 runs is shown in table 2.

	Best known value	HAS_QAP	ANTabu	ACO Solimanpur	ACO_GLS	Time (s)
Els 19	17212548	0.923	0	0	0	4
Tai 20b	122455319	0.243	0	0	0	5
Chr 25a	3796	3.0822	0.8957	0	0	3
Bur 26a	5426670	0	0	0	0	35
Bur 26b	3817852	0	0.0169	0	0	34
Bur 26c	5426795	0	0	0	0	34
Bur 26d	3821225	0	0	0	0	35
Bur 26e	5386859	0	0	0	0	34
Bur 26f	3782044	0	0	0	0	34
Bur 26g	10117172	0	0	0	0	33
Kra30a	88900	0.6299	0.2677	0	0	35
Kra 30b	91420	0.0711	0	0.0153	0	19
Nug30	6124	0.098	0	0.013	0	3

^aValues indicate the average gap between solution value and best known value in percent

Table 2. Compare results on QAP instances selected from QAPLIB (best results are in boldface)^a

Table 3 proves that for the instances with up to 30 tasks, ACO_GLS performs better than all other algorithms in comparison.

In order to generalize the application of our algorithm, large instances from the QAPLIB were studied with different classes of problems. Results are shown in Table 3. We have compared those algorithms on a set of 12 instances, ranging from 35 to 128 locations.

For larger instances, the results given by ANTabu are a little bit better, so we may have to perform more complicated local search in order to escape local minima in the problems with large instances. It is shown (table 3) that our algorithm ACO_GLS performs better than HAS-QAP. However, our algorithm can still obtain satisfactory solutions for large instance.

The proposed ACO-GLS algorithm proved to converge perfectly for instances up to 40 locations as shown in tables 2 and 3. This performance is quite satisfactory for industrial problems because real life problems usually do not exceed 30 to 40 locations. Therefore, this algorithm will be a very useful tool for layout optimization in the real life industrial case explained in this paper.

	Best known value	HAS_QAP	ANTabu	ACO_GLS	Time (s)
Tai 35a	2422002	1.762	0.215	0	109
Tai 35b	283315445	0.343	0.0408	0	112
Tai 40a	3139370	1.989	0.442	0	204
Tai 50a	4941410	2.800	0.781	1,28	228
Tai 60a	7208572	3.070	0.919	1.25	342
Tai 80a	13557864	0.663	0.663	1.53	1524
Wil 50	48816	0.061	0.008	0.01	1197
Sko42	15812	0.076	0	0	82
Sko49	23410	0.141	0.038	0.10	105
Sko56	34524	0.101	0.002	0.19	294
Sko64	48498	0.129	0.001	0.008	522
Esc 128	64	-	0	0	1292

Table 3. Compare results on QAP instances selected from QAPLIB (best results are in boldface)

5. Application to an industrial case

Our study (Hani et al., 2006; Hani et al., 2007) concerns an industrial layout problem for a train maintenance facility of the French railway system (SNCF).

The train maintenance facility is composed of buildings established on parallel rail tracks [15]. The cars to be treated arrive in batches and travel in the various buildings according to the sequence of their operations. They may travel transversally carried from one building to another by a transport which moves on a fixed trajectory. An on-rail transport permits movement along the rails. Some tasks require long processing, which would occupy their locations for a long time. These tasks represent bottlenecks for the facility.

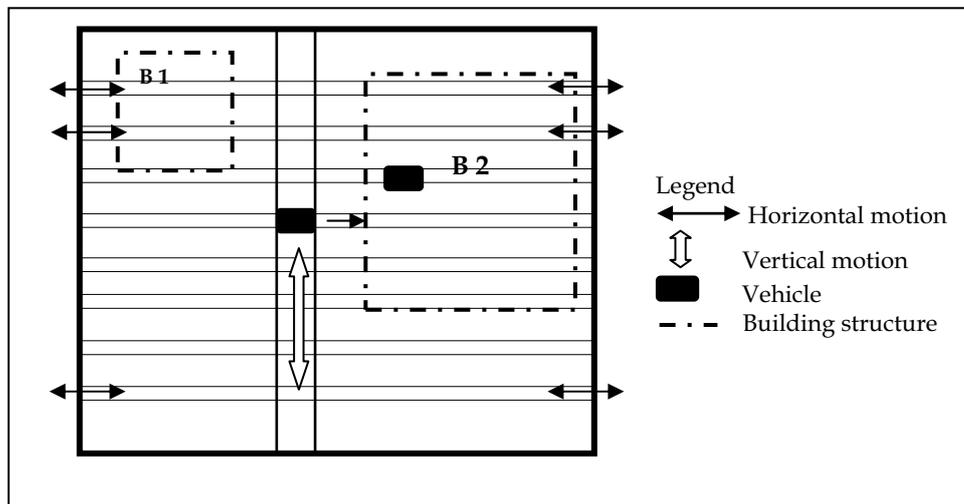


Figure 1. One example structure

In the current application, and due to the paucity of locations, some cars must be moved out of its building in order to give access to other cars in need of repair. The current facility layout has proved to be quite constraining for the production planning of the repair line. The problem is to find a layout of the resources in individual buildings in order to optimize the flow of cars between the buildings.

Figure 2 shows an example of a building composed of two parallel tracks and 6 locations. Access to the building is possible only via the lateral side. Suppose a car needs to pass from outside to location 2, and then to location 6. Then it is necessary to move the car which occupies location 1 or location 3 in order to let the new car access location 2. Then, in order to go to location 6, it is necessary to move the car on location 3 or that on location 1.

In other words, the problem is to find a new configuration of the resources in one of the buildings (figure 1) in order to optimize (minimize) the flow among all resources (facilities).

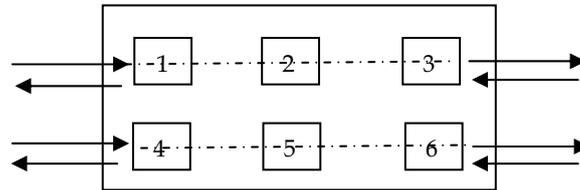


Figure 2. One building example

In order to model our problem, each rail is decomposed into zones called *car locations* where the maintenance tasks are performed.

The car locations can be categorized into three types:

- If a location is cluttered (e.g., stocks, workbench, etc.), then it is called *unusable*.
- If a location is occupied by a fixed resource (such as big machines for electric tests), it is called *specialized* since these resources cannot be moved.
- If a location is neither *unusable* nor *specialized*, it is said to be *standardized*.

Note that when unusable locations exist, access becomes even more difficult, as for example having location 6 as unusable in figure 2.

Outside the building, there exists a transport system which carries cars from one side to another of the facility. There are three transporters: two transversal and one on-rail transport that effects the movement of the cars on the rails.

In other words, the problem is to find a new resource configuration in one of the buildings (figure 2) in order to optimize (minimize) the production flow between all resources (facilities).

We consider N resources to be assigned to N sites or car locations in the building. Given a distance matrix D , where each element $d_{k,w}$ denotes a distance between location k and w , for $k, w = 1, 2, \dots, N$, a flow matrix F , where each element f_{ij} denotes a flow cost between resource i and j , for $i, j = 1, 2, \dots, N$.

The flow cost depends on the number of trips between two resources in a given time horizon. In the problem considered, the matrix flow is not symmetric because of precedence constraints.

The distance matrix is symmetric. The distance calculation is related to the minimum vehicle number to move inside a building in order to make an exchange. As an example in figure 1, $d(2,3) = 0$, $d(1,5) = 1$ (by crossing position 4) et $d(2,5) = 2$ (by crossing position 1 and 4).

Our problem is modeled as a QAP. The flow cost depends on the number of trips between two resources in a given time horizon. In the problem considered, the matrix flow is not symmetric because of precedence constraints.

The distance matrix is symmetric. The distance calculation is related to the minimum number of vehicles to move inside a building in order to make an exchange.

Model parameters:

N : total number of locations

r_{ij} : Resource j assigned to task i

$D_{k,w}$: Distance between locations k and w . This distance is defined as the number of usable locations between both resources.

$f_{r_{ij},r_{i',j'}}$: Production flow between the resources r_{ij} and $r_{i'j'}$. This flow is evaluated as the number of cars passing between the two resources.

$$p_{r_{ij},k} = \begin{cases} 1 & \text{if } r_{ij} \text{ is assigned to the location } k \\ 0 & \text{else} \end{cases} \quad (11)$$

$$TE_k = \begin{cases} 1 & \text{if location } k \text{ is standardized} \\ 0 & \text{else} \end{cases} \quad (12)$$

$$TES_k = \begin{cases} 1 & \text{if location } k \text{ is specialized} \\ 0 & \text{else} \end{cases} \quad (13)$$

In order to optimize the production flow, we define a quadratic function Z to minimize:

$$Z = \sum_i \sum_j \sum_{i'} \sum_{j'} \sum_k \sum_w f_{r_{ij},r_{i'j'}} \times D_{k,w} \times P_{r_{ij},k} \times P_{r_{i'j'},w} \quad (14)$$

If the unusable locations are excluded, the following constraints should be added:

$$\forall k: \sum_i \sum_j p_{r_{ij},k} = 1 \quad (15)$$

$$\forall i, j: \sum_k p_{r_{ij},k} = 1 \quad (16)$$

$$TE_k + TES_k = 1 \quad (17)$$

Constraints (15) and (16) are the standard constraints for the regular assignment problem. Constraint (17) implies that all occupied locations are either specialized or standardized.

The industrial problem consists of 72 locations with 27 unusable, 39 specialized and 6 standardized locations. 15 tasks with total of 27 resources had to be assigned. The actual resources assignment was taken as an initial condition for the algorithm. All calculations were done based upon data for one year planning.

As previously stated, the application of the hybrid GA algorithm to the industrial case has to exclude unusable and specialized locations from the locations choice; only standardized locations are considered.

The actual layout in the workshop produces a cost of 425, however, our algorithm ACO_GLS produces a solution with an improvement of 19.6% with respect to the actual layout. This means that it converges to a better solution, which proves its ability to solve an industrial layout problem.

We also found the exact solution of the problem by using an enumeration method since only six tasks needed to be assigned. The solution is the same as what was found by the algorithm. This implies that the algorithm converges to the optimal solution for this industrial problem.

The proposed application may be useful for the industrial case in the future. In fact, as stated above in the problem description, the industry is trying to increase its performance which means solving other facility problems. In addition, other vehicle sequences will be added, and many locations need to become free in order to accept new tasks. As it can be imagined, the future problem in the industry is to layout a greater number of locations which may reach 30 to 40 locations. The proposed ACO_GLS needs to be tested for large instance problems and its performance has to be evaluated with respect to other known algorithms. For this purpose, public sequences were tested and results were compared with other studies.

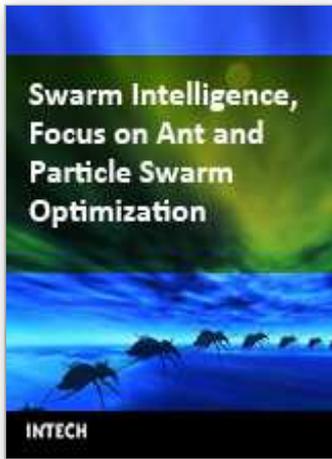
6. Conclusion

We have proposed a robust meta-heuristic algorithm for the layout problem modelled as a QAP. The algorithm is based on ant colony algorithm combined with a guided local search, and it uses an augmented cost function in order to guide the local search out of a local optimum. The performance of the proposed algorithm was also evaluated on a number of benchmark problems selected from the literature and compared with other heuristics developed for the facility layout problem as well as other algorithms recently developed for the QAP. The experimental results reveal that the proposed algorithm is effective and efficient for the facility layout problem considered. Other heuristic algorithms for the FLP shall be devised, tested, and compared with our algorithm in future studies.

7. References

- Baykasoğlu A., Gindy N.N.Z. (2001). A simulated annealing algorithm for dynamic layout Problem, *Computers & Operations research* 28, 1403-1426.
- Burkard R. E., Karisch S. & Rendel F. (1991). QAPLIB – A Quadratic Assignment Problem Library, *European Journal of Operational Research* 55, 115-119. Electronic update: <http://fmtbhpl.tu-graz.ac.at/~karisch/qaplib>.
- Caccetta L. & Kusumah Y.S. (2001). Computational aspects of the facility layout design problem, *Non-Linear Analysis* 47, 5599-5610.
- Chiang W.C. & Chiang C. (1998). Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation, *European Journal of Operational Research* 106, 457-488.
- D. Corne, M. Dorigo & F. Glover (1999). *New Ideas in Optimization*, Mac Graw Hill .
- M. Dorigo, E. Bonabeau & G. Theraulaz (2000) Ant algorithms and stimergy, *Future Generation Computer Systems* 2000; 16; 851-871.
- L. M. Gambardella, E. D. Taillard & M. Dorigo (1997) Ant colonies for the QAP, *Technical report IDISIA*; 4-97.

- Hani Y., Amodeo L., Yalaoui F. & Chen H. (2006). A hybrid genetic algorithm for solving an industrial layout problem, *Journal of Operation and Logistics*, 1, 4.1-4.11.
- Hani Y., Amodeo L., Yalaoui F. & Chen H. (2007) « Ant colony optimization for solving an industrial layout problem» *EJOR European Journal of Operational Research* 183, Issue 2, 633-642.
- Hicks C. (2004). A genetic algorithm tool designing manufacturing facilities in the capital goods industry, *International Journal of Production Economics* 90 (2), 199-211.
- Kusiak, S. & Heragu, S. (1987). The facility layout problem, *European Journal of Operational research*, 29, 229-251.
- Lee Y.H. & Lee M.H (2002): A shape-based block layout approach to facility layout problems using hybrid genetic algorithm, *Computers & Industrial engineering* 42, 237-248.
- V. Maniezzo, A. Colomi, & M. Dorigo (1994), The ant system applied to the quadratic assignment problem. *Technical report IRIDIA/, Université Libre de Bruxelles*, 94-28.
- V. Maniezzo, (1999) Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem, *INFORMS Journal on Computing*;11; 358-369.
- Martens J. (2004) Two genetic algorithmes to solve a layout problem in fashion industry. *European Journal of Operational Research* 154 (1), 304-322.
- Meller R. D., Narayanan V., Vance P.H. (1999) Optimal facility layout design, *Operations Research Letters* 23, 117-127.
- Mills P., Tsang E., Ford J. (2003). Applying an extended Guided Local Search to the Quadratic Assignment Problem, *Annals of Operational Research*, 118, 121-135.
- Sahni S. & Gonzales T. (1976) P-complete approximation problems, *Journal of Associated computer Machinery* 23 (5), 555-565.
- Solimanpur M, Vrat P. & Shankar R. (2004) Ant Colony optimization algorithm to the inter-cell layout problem in cellular manufacturing, *European Journal of Operational Research* 157, 592-606.
- T. Stützle, & M. Dorigo, (1999) Aco algorithm for the quadratic assignment problem, *Technical report IRIDIA/, Université Libre de Bruxelles*, 99-2.
- T. Stützle, & H. Hoos, (2000) MAX-MIN ant system, *Future Generation Computer Systems* 16 889-914.
- E. G. Talbi, O.Roux, C. Fonlupt & D. Robillard, (2001) Parallel Ant Colonies for the quadratic assignment problem, *Future Generation Computer Systems* 17, 441-449.



Swarm Intelligence, Focus on Ant and Particle Swarm Optimization

Edited by Felix T.S. Chan and Manoj Kumar Tiwari

ISBN 978-3-902613-09-7

Hard cover, 532 pages

Publisher I-Tech Education and Publishing

Published online 01, December, 2007

Published in print edition December, 2007

In the era of globalisation, the emerging technologies are governing engineering industries to a multifaceted state. The escalating complexity has demanded researchers to find the possible ways of easing the solution of the problems. This has motivated the researchers to grasp ideas from nature and implant them in the engineering sciences. This way of thinking led to the emergence of many biologically inspired algorithms that have proven to be efficient in handling computationally complex problems with competence, such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), etc. Motivated by the capability of the biologically inspired algorithms, the present book on "Swarm Intelligence: Focus on Ant and Particle Swarm Optimization" aims to present recent developments and applications concerning optimization with swarm intelligence techniques. The papers selected for this book comprise a cross-section of topics that reflect a variety of perspectives and disciplinary backgrounds. In addition to the introduction of new concepts of swarm intelligence, this book also presented some selected representative case studies covering power plant maintenance scheduling; geotechnical engineering; design and machining tolerances; layout problems; manufacturing process plan; job-shop scheduling; structural design; environmental dispatching problems; wireless communication; water distribution systems; multi-plant supply chain; fault diagnosis of airplane engines; and process scheduling. I believe these 27 chapters presented in this book adequately reflect these topics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yasmina Hani, Lionel Amodeo, Farouk Yalaoui and Haoxun Chen (2007). Hybrid Optimisation Method for the Facility Layout Problem, *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, Felix T.S. Chan and Manoj Kumar Tiwari (Ed.), ISBN: 978-3-902613-09-7, InTech, Available from:
http://www.intechopen.com/books/swarm_intelligence_focus_on_ant_and_particle_swarm_optimization/hybrid_optimisation_method_for_the_facility_layout_problem

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.