



Discrete Optimization

Ant colony optimization for solving an industrial layout problem

Y. Hani *, L. Amodeo, F. Yalaoui, H. Chen

ISTIT-OSI, (CNRS 2732) Université de Technologie de Troyes, 12 rue Marie Curie BP2060, 10010 Troyes cedex, France

Received 26 August 2005; accepted 6 October 2006

Available online 12 January 2007

Abstract

This paper presents ACO_GLS, a hybrid ant colony optimization approach coupled with a guided local search, applied to a layout problem. ACO_GLS is applied to an industrial case, in a train maintenance facility of the French railway system (SNCF). Results show that an improvement of near 20% is achieved with respect to the actual layout. Since the problem is modeled as a quadratic assignment problem (QAP), we compared our approach with some of the best heuristics available for this problem. Experimental results show that ACO_GLS performs better for small instances, while its performance is still satisfactory for large instances.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Layout problem; Quadratic assignment problem; Ant colony optimization; Guided local search

1. Introduction

The Facility Layout Problem (FLP) is to find a good configuration for machines, equipments or other resources in a given facility in order to optimize production flows while minimizing the total cost. It has a significant implication on the performance of a manufacturing system. There exist many applications for FLP such as workshop organization, construction of new production units, or equipment assignment. A full description of layout problem can be found in (Kusiak and Heragu,

1987). Layout problem is known to be NP-Hard (Sahni and Gonzales, 1976) and could be found in many classical and theoretical studies. However, only few layout industrial cases were treated in the literature. Hicks (2006) developed a genetic algorithm for minimizing material movement in a manufacturing cell with application on practical problems related to the capital good industry, Lee et al. (2005) proposed a genetic algorithm for solving multi-floor facility layout problems with the inner structure consisting of walls and passages. A study related to the fashion industry was presented by Martens (2004).

An extensive amount research has been conducted on the FLP; much of it is based on the Quadratic Assignment Problem (QAP). Other formulations exist such as mixed integer programming (Montreuil and Laforge, 1990; Montreuil

* Corresponding author. Tel.: +33 25 71 8091; fax: +33 25 71 5649.

E-mail addresses: yasmina.hani@utt.fr (Y. Hani), lionel.amodeo@utt.fr (L. Amodeo), farouk.yalaoui@utt.fr (F. Yalaoui), haoxun.chen@utt.fr (H. Chen).

et al., 1993; Meller et al., 1999) and graph theoretic model (Caccetta and Kusumah, 2001). Many methods used to solve the layout problem are essentially based on metaheuristics such as Genetic Algorithms (Tam, 1992; Tavakkoli-Moghaddain and Shayan, 1998; Lee et al., 2005), tabu search (Chiang and Chiang, 1998), simulated annealing (Baykasoğlu et al., 2001; Chwif et al., 1998; Mir and Imaam, 2001) and ant colony (Solimanpur et al., 2004; Solimanpur et al., 2005). Other methods combine simulated annealing with genetic algorithm such as FACOPT (Balakrishnan et al., 2003) and CRAFT developed by Armour et al. (1964). In order to ensure the convergence to a globally optimal solution with a minimum computation time, metaheuristics often include local search methods such 2-opt search (Lin, 1965). Another method known as Guided Local Search (GLS) (Voudouris, 1997) sits on the top of a local search and permits to escape from a local minima, and thus to ensure the global convergence. GLS has been successfully applied to the Traveling Salesman Problem (TSP) (Voudouris, 1999) and the QAP problem (Mills et al., 2003).

Ant colony optimization (ACO) is a method widely used for solving quadratic assignment problem. The first application was proposed by Maniezzo et al. (1994). Since that, many applications were proposed, and the differences were in the generation of solutions, the local search method and the pheromone updating. Stützle and Dorigo (1999) 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 Stützle and 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 (Talbi et al., 2001) using an ant colony optimization with a tabu local search procedure, was successfully applied to the QAP for large instances (i.e. 256 resources).

This paper proposes a method for solving a facility layout problems modeled as a QAP. It is based upon ant colony optimization with a GLS procedure to escape from local minima. The method is first applied to a particular industrial problem, and then, the performance is evaluated on small instances as well as large instances from the public library QAPLIB (Burkard et al. 1991). The results of our method are compared with those of Solimanpur et al. (2004), Gambardella et al. (1997) and Talbi et al. (2001).

The remainder of this paper is organized in four sections. Section 2 describes the facility layout problem and the industrial case modeled as a QAP. Section 3 presents the proposed ant algorithm, as well as the guided local search procedure for the QAP. Section 4 and 5 show modeling and results for the industrial problem and evaluates the proposed method's performance on some QAPLIB instances. Finally, Section 5 concludes the paper.

2. Problem description and formulation

2.1. Description

The problem comes from a train maintenance facility which is composed of buildings established on parallel railways. Each vehicle essentially crosses two buildings, x_1 and x_2 specialized in painting and disassembling respectively as shown in Fig. 1. Vehicles are first treated on external rails, and then move inside and between the two buildings following a given sequence before ending their course.

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

The cars to be treated arrive in batches and travel in the various buildings according to their sequences. They may travel transversally carried by a transporter which moves on a fixed trajectory. An on-rail transporter permits movements parallel to the rails. Some tasks require a long duration, which could occupy locations for a long time. These tasks represent bottlenecks for the workshop. In the current workshop, some cars must be moved out constantly in order to let access to other cars due to lack of loca-

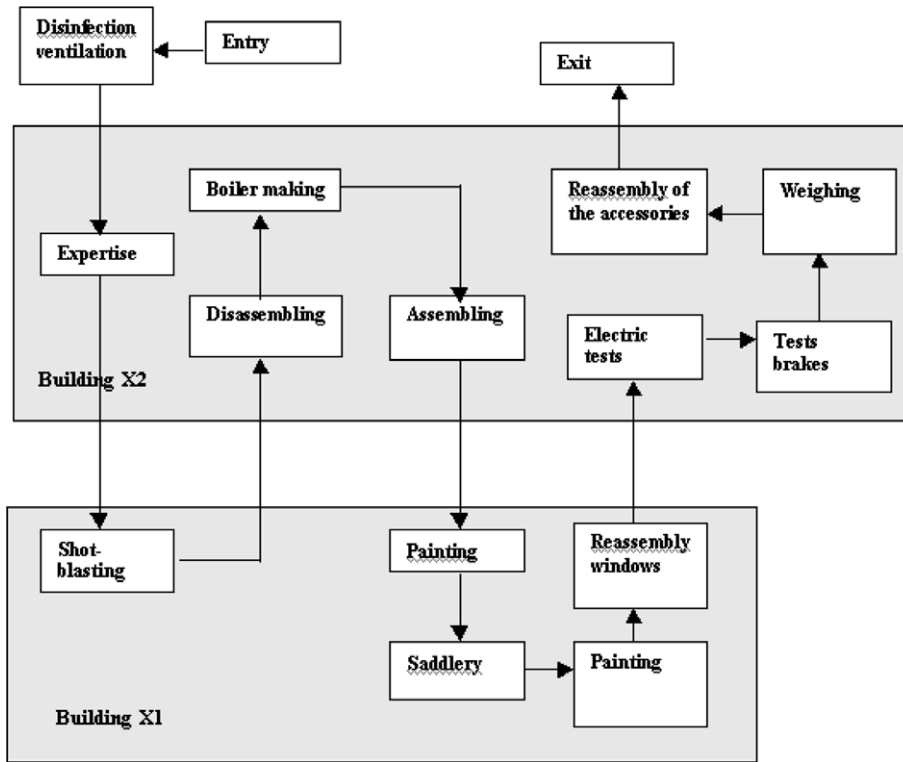


Fig. 1. The vehicle path in a train maintenance facility.

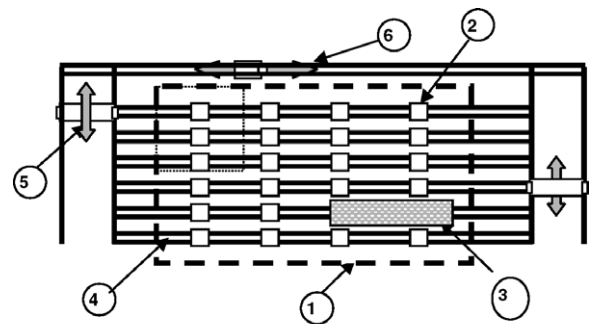
tions. The current workshop layout has been proved to be very constraining for the planning of the production line. The problem is to find a layout of the resources in one of the buildings, in order to optimize the flow of production between them.

In other words, the problem is to find a new resource configuration in one of the buildings (Fig. 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 $f_{i,j}$ 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 Fig. 3, $d(1,3) = 0$, $d(1,3) = 1$ (by crossing position 2) and $d(2,6) = 2$ (by crossing position 1 and 5).



- Legend:**
 1: Building area
 2: Car location
 3: Unusable location
 4: Circulation path
 5: Transversal transporter
 6: on-rail transporter

Fig. 2. Building to layout.

2.2. Quadratic assignment problem (QAP) formulation

The QAP has been traditionally used to model the FLP with some assumptions. In our industrial problem, the sizes of vehicle locations are identical and the distances between the locations are

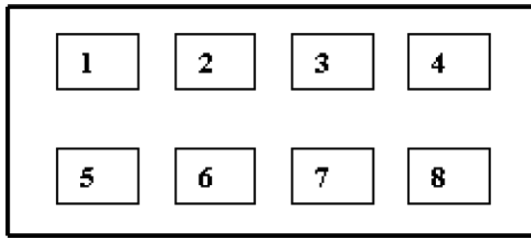


Fig. 3. An example of distance calculation inside a building.

predefined numeric values. Therefore, it was possible to formulate our problem as a QAP.

QAP which was first introduced by Koopmans and Beckman (1957) is the problem of allocating a set of facilities to a set of locations to minimize the total cost associated not only with the distances between locations, but also with the flows.

$f_{i,j}$ denotes the flow between facilities i and j and $d_{k,w}$ the distance between locations k and w . A variable $P(i,j)$ is then defined as:

$$P(i,j) = \begin{cases} 1 & \text{if the facility } i \text{ is assigned to the location } j, \\ 0 & \text{else.} \end{cases}$$

For most cases, the cost associated with facility i_1 assigned to location j_1 and facility i_2 assigned to location j_2 is considered as proportional to the product of the flow $f_{i_1 i_2}$ and the distance $d_{j_1 j_2}$. Thus, the QAP can be written as follows:

$$\begin{aligned} \text{Minimize } Z &= \sum_{i_1} \sum_{i_2} \sum_{j_1} \sum_{j_2} f_{i_1 i_2} d_{j_1 j_2} P(i_1, j_1) P(i_2, j_2) \\ \text{s.t. } &\forall j, \sum_i P(i, j) = 1, \quad \forall i, \sum_j P(i, j) = 1. \end{aligned}$$

After modeling of the problem we apply a method based on ant colony optimization to solve it.

3. Ant colony optimization and guided local search

In this section, we first present the ant colony optimization and the general algorithm. Then, we detail the elements of the ant colony algorithm adapted to the layout problem. We coupled ant algorithm with a guided local search. The definition of this method and the application on the Quadratic assignment problem are presented in Section 3.2. and the complete algorithm ACO_GLS is presented in Section 3.3.

3.1. Ant colony optimization

The principle of ACO algorithms (Corne et al., 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.

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

1. construction of solutions,
2. heuristic information,
3. pheromone updating,
4. selection probability,
5. local search,
6. diversification.

3.1.1. 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.

3.1.2. 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.

3.1.3. 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,

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

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

3.1.4. Selection probability

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

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

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 .

3.1.5. 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:

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

This algebraic simplification was used by Gambardella et al. (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.

3.1.6. Diversification

Used by Gambardella et al. (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

- (a) assign tasks to locations with a probability p ,
- (b) update the pheromones,
- (c) if the best solution is not improved until max_iter iterations, $\tau_{il} = 0$, except for the best solution,

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

3.2. Guided local search (GLS)

Guided Local Search (GLS) (Mills et al., 2003) is a metaheuristic which sits on the top of a 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 follows:

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

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 $s' =$ 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.

3.3. Complete algorithm

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. Application to the industrial case

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 $k, w = 1, 2, \dots, N$ and a flow matrix F , where each element $f_{i,j}$ denotes a flow cost between resource i and j , for $i, j = 1, 2, \dots, N$.

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

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

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

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

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.

5. Experimental results

5.1. Parameters

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

Table 1
Parameter values

Parameter	Value
AN	20
Alpha	0.6
<i>max_iter</i>	10
τ_0	0

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.

5.2. Industrial application

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

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 30–40 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

instances were tested and results were compared with other studies.

5.3. Generalization

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 3 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.

Table 2 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 30–40 locations. Therefore, this algorithm will be a very useful tool for layout optimization in the real life industrial case explained in this paper.

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

	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 26 ^e	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

^a Values indicate the average gap between solution value and best known value in percent.

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

	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

6. Conclusion

In this paper we have proposed a robust meta-heuristic algorithm for the layout problem modeled as a QAP. The algorithm is based on ant colonies optimization and guided local search. GLS uses an augmented cost function in order to guide the local search out of a local optimum. The development of this algorithm is motivated by an industrial case in a train maintenance facility. We applied it to one industrial case with six locations, and found the optimal solution. Since the number of locations is to be increased in the future, the performance of our proposed ant algorithm is tested over a number of problems selected from the literature and compared to many other existing algorithms. Results show that an improvement in performance compared to HAS-QAP, ANTabu and a Solimanpur et al. ACO for problems with the number of locations up to 40 can be obtained for our algorithm

ACO_GLS. Therefore, ACO_GLS is the most adapted algorithm for this industrial case and for other adapted layout problems with number of locations less than 40; however, results are still satisfactory for layout problems with larger instances.

The future work includes finding more complicated local search in order to have better results for large instances and finding a method that can treat problems with more complicated constraints.

References

- Armour, G.C., Buffa, E.S., Vollmann, T.E., 1964. Allocating facilities with CRAFT. *Harvard Business Review* 42, 136–158.
- Balakrishnan, J., Cheng, C-H., Wong, K-F., 2003. FACOPT: a user friendly FACility layout OPTimization system. *Comput Operat Res* 30, 1625–1641.
- Baykasoglu, 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://fmtbhl.tu-graz.ac.at/~karisch/qaplib>.
- Caccetta, L., Kusumah, Y.S., 2001. Computational aspects of the facility layout design problem. *Non-Linear Analysis* 7, 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.
- Chwif, L., Barretto, M.R.P., Moscato, L.A., 1998. A solution to the facility layout problem using simulated annealing. *Computers in industry* 36, 125–132.
- Corne, D., Dorigo, M., Glover, F., 1999. *New Ideas in Optimization*. Mac Graw Hill.
- Dorigo, M., Bonabeau, E., Theraulaz, G., 2000. Ant algorithms and stimergy. *Future Generation Computer Systems* 16, 851–871.
- Gambardella, L.M., Taillard, E.D., Dorigo, M. Ant colonies for the QAP, Technical report IDISIA, 1997, pp. 4–97.
- Hicks, 2006. A genetic algorithm tool designing manufacturing facilities in the capital goods industry, *International Journal of Production Economics*.
- Koopmans, T.C., Beckman, M., 1957. Assignment problems and the location of economic activities. *Econometrica* 25, 53–76.
- Kusiak, Heragu, S.S., 1987. The facility layout problem. *European Journal of Operational research* 29, 229–251.
- Lee, K.Y., Roh, M.I., Jeong, H.S., 2005. An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. *Computers & Operations Research* 32 (4), 879–899.
- Lin, S., 1965. Computer solutions for the traveling salesman problem. *Bell System Technology Journal* 44, 2245–2269.
- Maniezzo, V., Colorni, A., Dorigo, M. The ant system applied to the quadratic assignment problem. Technical report IRIDIA/, Université Libre de Bruxelles, 1994, pp. 94–128.
- Maniezzo, V., 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 algorithms to solve a layout problem in fashion industry. *European Journal of Operational Research* 54 (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 OR* 118, 121–135.
- Mir, M., Imaam, M.H., 2001. A hybrid optimization approach for layout design of unequal-area facilities. *Computers & Industrial Engineering* 39, 49–63.
- Montreuil, Laforge, A., 1990. A modeling framework for integrating layout design and flow network design. *Proceeding of Material Handling Research Colloquium*, 43–58.
- Montreuil, B., Venkatadri, U., Ratliff, H.D., 1993. Generating a layout from a design skeleton. *IIE Trans* 25, 3–15.
- Sahni, S., Gonzales, T., 1976. P-complete approximation problems. *Journal of Associated Computer Machinery* 23 (5), 555–565.
- Solimanpur, M., Vrat, M., Shankar, R., 2004. Ant Colony optimization algorithm to the inter-cell layout problem in cellular manufacturing. *European Journal of Operational Research* 157, 592–606.
- Solimanpur, M., Vrat, Prem, Shankar, Ravi, 2005. Ant algorithm for the single row layout problem in flexible manufacturing systems. *Computers & Operations Research* 32 (3), 583–598.
- Stützle, T., Dorigo, M. Aco algorithm for the quadratic assignment problem. Technical report IRIDIA/, Université Libre de Bruxelles, 1999, pp. 99–102.
- Stützle, T., Hoos, H., 2000. MAX-MIN ant system. *Future Generation Computer Systems* 16, 889–914.
- Talbi, E.G., Roux, O., Fonlupt, C., Robillard, D., 2001. Parallel ant colonies for the quadratic assignment problem. *Future Generation Computer Systems* 17, 441–449.
- Tam, K.A., 1992. Genetic algorithms, function optimization, and facility layout design. *European Journal of Operational Research* 63, 322–346.
- Tavakkoli-Moghaddain, R., Shayan, E., 1998. Facilities layout design by genetic algorithms. *Computers Industrial Engineering* 35, 527–530.
- Voudouris, Guided Local Search for Combinatorial Optimisation Problems, Ph.D. thesis, Department of Computer Science, University of Essex, 1997.
- Voudouris, E., Tsang, P.K., 1999. Guided Local Search and its application to the Travelling Salesman Problem. *European Journal of Operational Research* 113 (2), 469–499.