ANTabu — enhanced version
LIL-99-1

Olivier Roux*, Cyril Fonlupt*, El-Ghazali Talbi◇, and Denis Robilliard*

* LIL,
Université du Littoral,
BP 719, 62228 Calais, France

◇ LIFL,
Université de Lille 1, Bat. M3,
59655 Villeneuve d'Ascq Cedex, France

Abstract. Many methods currently used in combinatorial optimization are inspired by adaptative natural behaviors or natural systems, *e.g.* hill-climbing local search, genetic algorithms, simulated annealing, … These heuristics provide some very good results, even when the problem size makes it impossible to use more traditional exact methods (such as branch and bound, ...). Ants system algorithms belong to this class of adaptive or evolutionnary "nature inspired" algorithms, and are based on the natural behavior of ants. These algorithms have been applied successfully to many optimisation problems like travelling salesman (TSP), job shop scheduling (JSP), graph colouring, quadratic assignment problem (QAP), …
This paper presents ANTabu, an ants system algorithm for the QAP. We have designed ANTabu with a parallel model, thus allowing it to take advantage of the power of networks of workstations. The co-operation between simulated ants is provided by a pheromone matrix that plays the role of a global memory, with a refined pheromone update mechanism. The exploration of the search space is guided by the evolution of pheromones level, while exploitation has been boosted by a tabu local search. Special care has also been taken in the design of a diversification phase, based on a frequency matrix. We give results that have been obtained on benchmarks instances from the QAPlib[1]. We show that they compare favourably with other top of the art algorithms dedicated for the QAP.
Keywords : Ants Systems, Quadratic Assignment Problem, Tabu search, Parallelism, Pheromone trails, Global memory.

1 Introduction

1.1 The Quadratic Assignment Problem

The QAP is a combinatorial optimization problem, first described by Koopmans and Beckmann in [KB57]. It consists in finding the best assignment of n facilities to n locations at minimum cost. There are several QAP applications like

[1] http ://fmatbhp1.tu-graz.ac.at/%7Ekarisch/qaplib/

arrangement of departments in hospitals, minimization of the total wire length in electronic circuits...

This problem can be formally described as follows :

Given two $n \times n$ symetric matrices D and C, where D is the inter-city distance matrix and C the inter-city flow matrix, find a permutation π^* minimizing the objective function f :

$$\min_{p \in \Pi(n)} f(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} c_{p_i p_j}$$

where :

p is a permutation.
p_i is the i^{th} element of p.
$\Pi(n)$ is the set of all permutations of n elements.
$d_{i,j}$ is the distance between objects i and j (*i.e.* an element of D).
$c_{a,b}$ is the cost of flow between positions a and b (*i.e.* an element of C).

Thus a solution is an integer permutation, with an element of the permutation being an object number and its position within the permutation being the object location.

The QAP is a NP-hard problem and finding an ε approximation to this problem is known to be NP-complete [SG76], so problems of size larger than 20 are still considered as intractable for exact methods. Several heuristics have been proposed for finding near-optimum solutions to large instances, see [Con90], [SK90], [Tai91], [BT94], [FF94], [BPT96], [HTG96], [SV96], [CMMT97], [MF97], [GTD98], [THG97]. We compare our results with those from oome of these methods in the next sections of this paper.

1.2 Ants and ants systems

The ants based algorithms were introduced in Marco Dorigo's PhD [Dor92]. They are inspired by the observation that, using very simple communication mechanisms, an ant group is able to find the shortest path between any two points. During their trips a chemical trail (pheromone) is left on the ground. The role of this trail is to guide the other ants towards the target point.

For one ant, the path is chosen accordingly to the quantity of pheromone. Furthermore, this chemical substance has a decreasing action over time, and the quantity left by one ant depends on the amount of food found. As illustrated in Fig. 1, when facing an obstacle, there is an equal probability for every ant to choose the left or right path. As the left trail is shorter and so requires less travel time, it will end up with a higher level of pheromone after many ants passages thus being more and more attractive. This fact will be increased by pheromone evaporation.

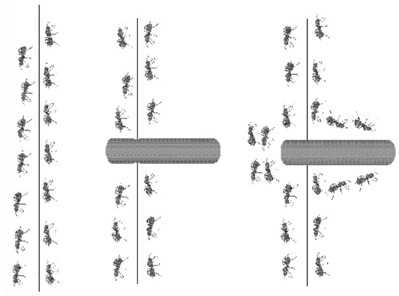

Fig. 1. Ants facing an obstacle

This subtle principle of communication has been used as a guideline for the design of heuristics dedicated to some combinatorial optimization problems, notably the TSP in [DMC96,GD95,DG96], vehicule routing problems [BHS97], load balancing in communication networks [CD97], real functions optimization [BP95], graph coloring problems [CH97], ... Several works have also been targeted at the QAP, for example Stüzle's Min-Max algorithm [Stü97], or Gamberdella *et al.*'s HAS-QAP [GTD98].

1.3 A brief overview of this paper

In the following sections, we will first introduce the ANTabu algorithm, stressing the original points that make it differ from previous typical ants systems. Then we will review numerical results obtained on a set of instances from the QAPlib [BKR97], whose sizes range from 20 up to 100. We compare our results with those from a wide set of publications in the field of meta-heuristics for the QAP. We discuss the difference in results for regular and irregular instances. At last, we conclude and propose possible ways for future works.

2 ANTabu

The main advantage of ants sytems is to simulate a global memory. It thus allows to build a knowledge base on already visited sites and this in a stochastic way. The more an element is found in good solutions, the more likely it is supposed to be part of the optimal solution. The evaporation phenomenon helps in avoiding premature convergence to a local optimum.

2.1 ANTabu scheme

ANTabu is based on the co-operation of two methods (the Ants system and the Tabu algorithm). The general outline of our method is presented in Fig. 2.

First of all, initialization of pheromone matrix is made with an arbitrary value τ_0 proportional to a good solution. This solution is obtained after optimization

```
1) Initializations
   1a) Generate m (ant number) permutations p_k with (1 ≤ k ≤ m), of
       length n
   1b) Pheromone matrix initialization
      1b1) apply local search on permutation
      1b2) Select the best solution  : p*
      1b3) Pheromone matrix initialization F
   1c) Initialization of frequency matrix to 0
   1d) intensification is TRUE
2) FOR i = 1 to I^max
   2a) FOR each permutation p_k (1 ≤ k ≤ m)
      2a1) n/3 guided swaps by pheromone matrix : result is p̂_k
      2a2) Apply Tabu search (p̂_k gives p̃_k)
   2b) IF each f(p̃_k) ≥ f(p_k)
            THEN intensification is FALSE
   2c) FOR each ants (k)
       IF intensification is TRUE AND f(p̃_k) > f(p_k)
            THEN solution comes back to p_k
            ELSE p_k = p̃_k
   2d) IF p̃_k < p* for at least one ant k
            THEN update the best solution found so far p* and
            intensification is TRUE
   2e) Update the pheromone matrix with all solutions
   2f) Update the fequency matrix
   2g) IF n/2 iterations since the last update of p*
            THEN diversification generate new solutions with
            frequency matrix
```

Fig. 2. ANTabu algorithm

of random solutions. During the run of the ANTabu algorithm, this matrix will be updated by solutions provided by the ants.

Each ant applies some swaps to its solution, these swaps are only guided by the pheromone matrix. This operation gives a preliminary solution. The local search method (Tabu algorithm) will be then applied to these solutions. Note that in our implementation we have restricted the Tabu method to a limited number of steps, to avoid premature convergence.

These solutions are then provided to the pheromone matrix update procedure. This is a two phases procedure:

- an evoparation scheme is applied (*i.e* the values of the pheromone matrix are decreased)
- each solution will be used to update the matrix (*i.e.* reinforcement of the assignements found in solutions), but see details in the next subsection.

At this step, ANTabu may either enter in an intensification phase if the best previous solution is improved, or in a diversification phase if there is no improvement after some iterations.

2.2 A closer look at global memory management

Most ants algorithms only take into account the best solution found so far for updating the pheromone matrix. We have devised a new strategy where each ant adds a contribution proportional to its fitness value (the better the solution, the higher the contribution). This contribution is weakened by dividing the difference between the solution and the worst one with the best one. So far, the new update formulae are :

– for all matrice values :

$$\forall i, j \in [1, n], \tau_{i,j}^{g} = (1 - \alpha)\tau_{i,j}^{g-1}$$

– for all solutions p of one iteration :

$$p \in P^g, \forall i, j \in [1, n], \tau_{i,j}^{g} = \tau_{i,j}^{g} + \frac{\alpha}{f(p)} \frac{f(p^-) - f(p)}{f(p^*)}$$

f is the objective function.
i is the position in the solution and j is the value.
$\tau_{i,j}^{g}$ is the pheromone trail value at the g^{th} iteration in position i, j.
p^- is the worst solution found so far.
p^* the best one.
P^g is the set of solutions at the g^{th} iteration.

The diversification strategy of the ANTabu algorithm tries to focus on unexplored areas. Instead of re-initializing the pheromone matrix and randomly generating new solutions, ANTabu uses a *frequency matrix*. This matrix holds frequencies for all previous assignments, and is used when a diversification phase is triggered. During the diversification phase, least chosen affectations are chosen to generate new solutions. This diversification scheme will force the ants system to start from totally new solutions with new structures.

2.3 Parallel implementation

The ANTabu has been parallelized for heterogeneous networks, each ant being run by a host when possible. In order to take advantage of the parallel architecture, we have built an asynchronous program. In such a case, an idle host will be provided with a new ant to work with.

The master receives informations from ants. Based on these results, pheromone matrix and frequency matrix are updated asynchronously.

Slaves/ants receive a new solution from the master with the updated pheromone matrix. For each ant, an iteration consists in building a new solution, improving it using the Tabu search method and sending it back to the master. Note that in our asynchronous implementation no host stays idle for a long time.

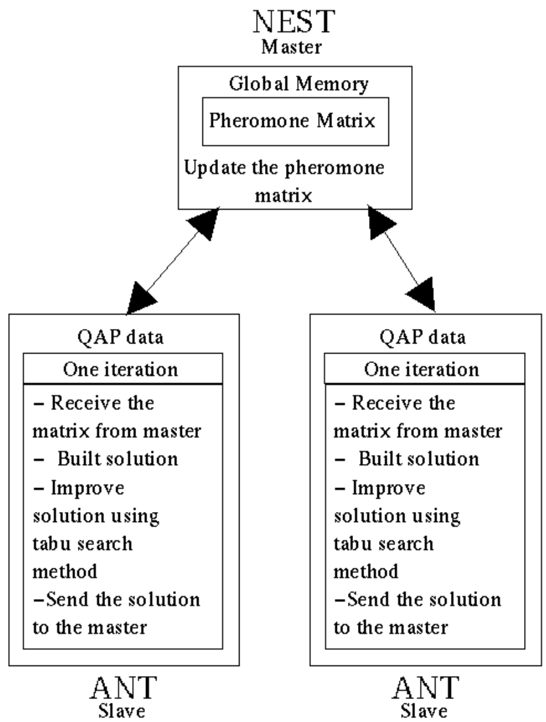

Fig. 3. Parallel model

3 Numerical results

In all our experiments, we have been working with a population of 10 ants. The experiments found in the first two tables (see Tab. 1 and Tab. 2) were run on a network of 10 Silicon Graphics Indy workstations. The other results in Tab. 3 and Tab. 4 were run on a network composed of 3 PC (PII 300). Unless otherwise stated, the values we give are averaged on ten independent executions.

We first compare ANTabu with HAS-QAP [GTD97], an ants system, and then we compare it with PATS [THG97], a parallel Tabu search. These two algorithms are known to provide top of the art results for the QAP, so this allows us to discuss the relative gains brought on one hand from ants cooperation, on the other hand from Tabu local search. Then we extend the range of competitors to a wider set of methods.

When comparing with other heuristics, the reader should notice that it is almost impossible to use the same experimental setting (*e.g.* machine, machine load, compiler efficiency, algorithm coding, . . .) than other authors. Nonetheless, in order to provide a minimal guideline for comparison, we have used the same computing time, when time data are available.

3.1 Comparisons with HAS-QAP and PATS

We have selected 12 problems from the QAPlib [BKR97] either from the irregular class problems (`bur26d, bur26b, chr25a, els19, kra30a, tai20b, tai35b`) or from the regular class problems (`nug30, sko42, sko64, tai25a` and `wil50`).

The parameters are : 10 iterations for each of our 10 ants (similar to those for HAS-QAP), with a Tabu search restricted to $5n$ iterations, and a pheromon evaporation rate of $\alpha = 0.1$. Results are shown in Tab. 1, those for HAS-QAP are taken straight from [GTD97], best figures are in boldface. The difference relatively to the QAPlib optimum is given as a percentage gap for both algorithms.

Table 1. Quality of HAS-QAP and ANTabu. Best results are in boldface.

Problem Name	Best known	HAS-QAP	ANTabu
`bur26b`	3817852	0.106	**0.018**
`bur26d`	3821225	0.002	**0.0002**
`chr25a`	3796	15.69	**0.047**
`els19`	17212548	0.923	**0**
`kra30a`	88900	1.664	**0.208**
`tai20b`	122455319	0.243	**0**
`tai35b`	283315445	0.343	**0.1333**
`nug30a`	6124	0.565	**0.029**
`sko42`	15812	0.654	**0.076**
`sko64`	48498	0.504	**0.156**
`tai25a`	1167256	2.527	**0.843**
`wil50`	48816	0.211	**0.066**

It is clear that ANTabu outperforms HAS-QAP. Nonetheless we may wonder if ants are of any use in improving the performance we expect from the Tabu part of our algorithm. This has led us to the following comparison with PATS [THG97], which consists in a set of Tabus running in a distributed fashion on a network of workstations (including Intel PCs, Sun and Alpha workstations). The load of each workstation is monitored and Tabus are automatically launched on idle machines and stopped on busy ones, using the MARS operating system layer [HTG96].

For this comparison, we have studied instances from two classes of problems :

- one instance with random uniform cost and distance matrices : `tai100a`
- a set of instances with random cost matrix : `sko100a, sko100b, sko100c, sko100d, wil100`

Results are shown in Table 2. Best found for PATS and ANTabu is the best solution out of ten runs. Running times are in minutes, and correspond to the mean execution time over 10 runs on a 126 machines network for PATS, and on a 10 machines network for ANTabu (thus the ANTabu algorithm was at a great disadvantage in this regard).

Table 2. Compared results from PATS and ANTabu. Best results are in boldface

	`tai100a`	`sko100a`	`sko100b`	`sko100c`	`sko100d`	`wil100`
QAPlib						
Best known	21125314	152002	153890	147862	149576	273038
PATS						
Best found	21193246	152036	153914	147862	149610	273074
Gap	0.322	0.022	0.016	0	0.023	0.013
Time	117	142	155	132	152	389
ANTabu						
Best found	21184062	152002	153890	147862	149578	273054
Gap	**0.278**	**0**	**0**	**0**	**0.001**	**0.006**
Time	139	137	139	137	201	139

Results show that the ANTabu system finds better results using less computing resources (see experimental setting in the above paragraph). Notice that best known solutions have been found in three out of four sko100 instances.

3.2 Comparisons with a wider set of algorithms

In this section ANTabu is compared with:

- the reactive tabu search (RTS) from Battiti and Tecchiolli [BT94]
- the tabu search (TT) from Taillard [Tai91]
- the genetic hybrid method (GH) from Fleurent and Ferland [FF94]
- the simulated annealing (SA) from Connolly [Con90]
- the HAS-QAP, previously cited, from Gambardella *et al.*, but with a computing time extended to 100 iterations for each of the 10 ants. Again, results are taken from [GTD98].

It has been shown in [Tai95] that these methods do not have the same effectiveness according to whether they are applied to so-called regular or irregular instances, where regular instances have a flow dominance lower than 1.2 as opposed to irregular ones which are also sometimes called "structured" instances.

Results for regular instances are shown in Tab. 3. It is clear that ANTabu dominates the scene.

Next we have compared those algorithms on a set of 20 irregular instances ranging from n=10 locations to n=80. Results are in Tab. 4. In this case best results are obtained by HAS-QAP, which found best average fitness for a set of sixteen instances. ANTabu found the best average for only thirteen instances.

Table 3. Compared results on regular instances with the same computing time. *Best results are in boldface. Values are the average of gap between solution value and best known value in percent over ten runs.*

Problem name	n	Best known value	TT	RTS	SA	GH	HAS-QAP	ANTabu	Seconds
nug20	20	2570	**0**	0.911	0.070	**0**	**0**	**0**	30
nug30	30	6124	0.032	0.872	0.121	0.007	0.098	**0**	83
sko42	42	15812	0.039	1.116	0.114	0.003	0.076	**0**	248
sko49	49	23386	0.062	0.978	0.133	0.040	0.141	**0.038**	415
sko56	56	34458	0.080	1.082	0.110	0.060	0.101	**0.002**	639
sko64	64	48498	0.064	0.861	0.095	0.092	0.129	**0.001**	974
sko72	72	66256	0.148	0.948	0.178	0.143	0.277	**0.074**	1415
sko81	81	90998	0.098	0.880	0.206	0.136	0.144	**0.048**	2041
sko90	90	115534	0.169	0.748	0.227	0.196	0.231	**0.105**	2825
tai20a	20	703482	0.211	0.246	0.716	0.268	0.675	**0**	26
tai25a	25	1167256	0.510	**0.345**	1.002	0.629	1.189	0.736	50
tai30a	30	1818146	0.340	0.286	0.907	0.439	1.311	**0.018**	87
tai35a	35	2422002	0.757	0.355	1.345	0.698	1.762	**0.215**	145
tai40a	40	3139370	1.006	0.623	1.307	0.884	1.989	**0.442**	224
tai50a	50	4941410	1.145	0.834	1.539	1.049	2.800	**0.781**	467
tai60a	60	7208572	1.270	**0.831**	1.395	1.159	3.070	0.919	820
tai80a	80	13557864	0.854	**0.467**	0.995	0.796	2.689	0.663	2045
wil50	50	48816	0.041	0.504	0.061	0.032	0.061	**0.008**	441

In an attempt to understand the reasons behind this difference in behavior between regular and irregular instances, we have also monitored not only average results but also best and worst ones. These data are shown in Tab. 5. Notice that an optimal solution is found at least once in ten runs for almost all problems. Thus we think that the slightly low average from Tab. 4 could be due to low quality local optima that are found after some unlucky diversification phase. It may well be the price to pay for a more complete exploration of the search space.

Table 4. Compared results on iregular instances with the same computing time. *Best results are in boldface. Values are the average of gap between solution value and best known value in percent over ten runs.*

Problem name	n	Best known value	TT	RTS	SA	GH	HAS-QAP	ANTabu	Seconds
bur26a	26	5426670	0.0004	-	0.1411	0.0120	**0**	**0**	50
bur26b	26	3817852	0.0032	-	0.1828	0.0219	**0**	0.0169	50
bur26c	26	5426795	0.0004	-	0.0742	**0**	**0**	**0**	50
bur26d	26	3821225	0.0015	-	0.0056	0.0002	**0**	**0**	50
bur26e	26	5386879	**0**	-	0.1238	**0**	**0**	**0**	50
bur26f	26	3782044	0.0007	-	0.1579	**0**	**0**	**0**	50
bur26g	26	10117172	0.0003	-	0.1688	**0**	**0**	**0**	50
bur26h	26	7098658	0.0027	-	0.1268	0.0003	**0**	**0**	50
chr25a	25	3796	6.9652	9.8894	12.4973	2.6923	3.0822	**0.8957**	40
els19	19	17212548	**0**	0.0899	18.5385	**0**	**0**	**0**	20
kra30a	30	88900	0.4702	2.0079	1.4657	**0.1338**	0.6299	0.2677	76
kra30b	30	91420	0.0591	0.7121	0.1947	0.0536	0.0711	**0**	86
tai20b	20	122455319	**0**	-	6.7298	**0**	0.0905	**0**	27
tai25b	25	344355646	0.0072	-	1.1215	**0**	**0**	**0**	50
tai30b	30	637117113	0.0547	-	4.4075	0.0003	**0**	**0**	90
tai35b	35	283315445	0.1777	-	3.1746	0.1067	**0.0256**	0.0408	147
tai40b	40	637250948	0.2082	-	4.5646	0.2109	**0**	0.4640	240
tai50b	50	458821517	0.2943	-	0.8107	0.2142	**0.1916**	0.2531	480
tai60b	60	608215054	0.3904	-	2.1373	0.2905	**0.0483**	0.2752	855
tai80b	80	818415043	1.4354	-	1.4386	0.8286	**0.6670**	0.7185	2073

4 Conclusions

In this paper we have proposed a powerful and robust algorithm for the QAP resolution, based on the hybridization of a Tabu search and an ants system. Compared with previous ants systems for the QAP, we have refined the ants cooperation mechanism, both in the pheromone matrix update phase and in the exploitation/diversification phase (using a frequency matrix). We have chosen the Tabu as a local search method, thus demonstrating that it is possible to

Table 5. Best and worst results on ten runs on irregular instances for ANTabu. *Values are gaps between solution value and best known value in percent. Notice that an optimal solution is found in all but one problems.*

Problem	Average	Seconds	Worst	Best
bur26a	0.0000	50	0.0000	0.0000
bur26b	0.0169	50	0.1693	0.0000
bur26c	0.0000	50	0.0000	0.0000
bur26d	0.0000	50	0.0000	0.0000
bur26e	0.0000	50	0.0000	0.0000
bur26f	0.0000	50	0.0000	0.0000
bur26g	0.0000	50	0.0000	0.0000
bur26h	0.0000	50	0.0000	0.0000
chr25a	0.8957	40	4.5838	0.0000
els19	0.0000	20	0.0000	0.0000
kra30a	0.2677	76	1.3386	0.0000
kra30b	0.0000	86	0.0000	0.0000
tai20b	0.0000	27	0.0000	0.0000
tai25b	0.0000	50	0.0000	0.0000
tai30b	0.0000	91	0.0000	0.0000
tai35b	0.0408	148	0.2212	0.0000
tai40b	0.4640	241	2.6239	0.0000
tai50b	0.2531	486	0.9075	0.0000
tai60b	0.2752	860	1.5608	0.0000
tai80b	0.7185	2091	1.8549	0.0019

use a powerful local search without being hindered by disproportionate overheads. Results show interesting performances, coming from the complementary gains brought by the mixed use of parallelism, efficient local search and ants-like cooperation. The comparison with the PATS algorithm pleads for a more widely spread use of inter-process communication in parallel heuristics. Future works include finding a better tuning of parameters and porting under MARS to allow runs on large heterogeneous networks. We are also interested in investigating more closely on the impact of the diversification phase, and in leading a finer study on the conditions needed for appearance of a fruitful cooperation in parallel agents systems.

References

[BHS97] B. Bullnheimer, R.F. Hartl, and C. Strauss. A new rank based version of the ant system: A computational study. Working paper, University of Vienna, Austria, 1997.

[BKR97] R.E. Burkard, S.E. Karisch, and F. Rendl. Qaplib - a quadratic assignment problem library. *Journal of Global Optimization-10*, pages 391–403, 1997.

[BP95] G. Bilchev and I. Parmee. The ant colony metaphor for searching continuous design spaces. *Lectures Notes in Computer Science 993*, pages 24–39, 1995. T. Fogarty (Ed.).

[BPT96] V. Bachelet, P. Preux, and E. G. Talbi. Hybrid parallel heuristics : Application to the quadratic assignment problem. In *Parallel Optimization Colloquium*, Versailles, France, Mar 1996. POC96.

[BT94] R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA J on Computing*, 6:126–140, 1994.

[CD97] G. Di Caro and M. Dorigo. Antnet: A mobile agents approach to adaptive routing. Technical Report IRIDIA/97-12, IRIDIA, Université Libre de Bruxelles, Belgium, 1997.

[CH97] D. Costa and A. Hertz. Ants can colour graphs. *295-305*, Journal of the Operational Research Society(48), 1997.

[CMMT97] V.-D. Cung, T. Mautor, P. Michelon, and A. Tavares. A scatter search based approach for the quadratic assignment problem. In *Proceedings of the IEEE Internatinnal Conference on Evolutionary Computation and Evolutionary Programming.*, pages 165–170, Indianapolis, USA, 1997. ICEC'97.

[Con90] D.T. Connolly. An improved annealing scheme for the qap. *Eur J Op Res 46*, pages 93–100, 1990.

[DG96] M. Dorigo and L.M. Gambardella. A study of some properties of ant-q. In *Proceedings of PPSN IV-Fourth -International Conference on Parallel Problem Solving From Nature*, pages 656–665, Berlin, Germany, September 22-27 1996.

[DMC96] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 1-Part B(26):29–41, 1996.

[Dor92] M. Dorigo. *Optimization, Learning and Natural Algorithms.* PhD thesis, Politecnico di Milano, Italy, 1992.

[FF94] C. Fleurent and J Ferland. Genetic hybrids for the quadratic assignment problem. *DIMACS Serie in Mathematics and Theoretical Computer Science*, 16, 1994.

[GD95] L.M. Gambardella and M. Dorigo. Ant-q: A reinforcement learning approach to the traveling salesman problem. In A. Prieditis and S. Russell (Eds.), editors, *Proceedings of ML-95 - Twelfth International Conference on Machine Learning*, pages 252–260, Tahoe City, CA, 1995. Morgan Kaufmann.

[GTD97] L.M. Gambardella, E. Taillard, and M. Dorigo. Ant colonies for the qap. Technical Report IDSIA 97-4, IDSIA, Lugano, Switzerland, 1997.

[GTD98] L.M. Gambardella, E. Taillard, and M. Dorigo. Ant colonies for the qap. *Accepted for publication in the Journal of the Operational Research Society*, 1998.

[HTG96] Z. Hafidi, E-G. Talbi, and J-M. Geib. Mars : Un ordonnanceur adaptatif d'applications parallèles dans un environnement multi-utilisateurs. In *RenPar'8-8ème Rencontres Francophones du Parallélisme*, pages 37–40, Bordeaux, France, Mai 1996.

[KB57] T.C. Koopmans and M.J. Beckmann. Assignment problems and localisation of activities. *Economica*, 25(53-76), 57.

[MF97] P Merz and B. Freileben. A genetics local search to the quadratic assignment problem. In *Internationnal Conference on Genetic Algorithms, ICGA'97*, pages 465–472, New Lancing, Michigan, USA, 1997.

[SG76] S. Sahni and T. Gonzales. P-complete approximation problems. *Journal of ACM-23*, pages 556–565, 1976.

[SK90] J. Skorin-Kapov. Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing*, Vol.2(no. 1):33–45, 1990.

[Stü97] T. Stützle. Max-min ant system for quadratic. Technical Report AIDA-97-04, AIDA, Darmstadt University of Technology, Computer Science Department, 1997.

[SV96] L. Sondergeld and S. Voß. *Meta-Heuristics : Theory and applications*, pages 489–502. Kluwer Academic Publishers : Boston/London/Dornecht, 1996.

[Tai91] E.D. Taillard. Robust taboo search for the quadratique assignment problem. *Parallel Computing 17*, 17:443–455, 1991.

[Tai95] E.D. Taillard. Comparison of iterative searches for the quadratic assignment problem. In *Location Science 3*, pages 87–105, 1995.

[THG97] E-G. Talbi, Z. Hafidi, and J-M. Geib. Parallel adaptive tabu search for large optimization problems. In *MIC'97-2nd Metaheuristics International Conference*, Sophia Antipolis, France, Juillet 1997.