AN ANT APPROACH FOR STRUCTURED QUADRATIC ASSIGNMENT PROBLEMS

Éric D. Taillard, Luca M. Gambardella
IDSIA, Corso Elvezia 36, CH-6900 Lugano, Switzerland.
Extended abstract IDSIA-22-97

ABSTRACT.

The paper presents a new heuristic method for the quadratic assignment problem. This method is based on an artificial ant system, hybridized with a fast local search. The method is easy to implement and numerical results show that it is fast and efficient for structured problems.

Key words : Quadratic assignment problem, ant system, adaptive memory programming.

1. INTRODUCTION.

Ant systems for finding heuristic solutions to combinatorial optimization problems have been proposed by Corloni, Dorigo and Maniezzo in 1991. The basic idea of ant systems is to imitate the behaviour of real ants when they are looking for food. An artificial ant built a solution of the problem to solve, being myopically guided by two informations : the objective function value (analogue to the presence of food for real ants) and traces left by other artificial ants having already built solutions (analogue of a chemical information left on the floor by real ants for guiding other ants). After having built a new solution, the artificial ant update the traces, taking into consideration the quality of the solution just built. The rôle of the traces is to implement an adaptive memory that evolves in time.

The quadratic assignment problem (QAP) is a NP-hard combinatorial optimization problem whose goal is to find a permutation π of n elements that minimizes the following objective function :

$$\min_{\pi \in P(n)} f(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \cdot b_{\pi_i \pi_j}$$

Where $P(n)$ is the set of permutations of n elements, (a_{ij}) and (b_{ij}) are two $n \times n$ matrices (data). This problem can be solved exactly for very small sizes only ($n \leq 25$). So, it is necessary to use heuristic methods to get good solutions for larger problems. The ant system presented in this paper is rather loosely connected to the behaviour of real ants but it is rather inspired by a successful approach to the travelling salesman problem due to Dorigo and Gambardella and based on an ant system. The approach presented in this paper is able to find out the structure of good solutions to the QAP, if such a structure exists.

2. THE ANT SYSTEM FOR QAP.

Very shortly, the process executed by an artificial ant searching for a permutation π can be sketched as follows :

1) *Initialize variable* r *and all the entries of the matrix of the traces,* τ*, to* r.
2) *For* k = *1 to* k^{max} *do :*
   - 2a) *Build a new solution* π*, using* τ*, in a probabilistic way.*
   - 2b) *Locally optimize* π*, to obtain an improved solution* μ*.*
   - 2c) *Update* π^**, the best solution found by the process and* τ*, using* μ *and* π^**.*

The elements τ_{ij} of the matrix of traces are measuring the preference or the interest of setting $\pi_i = j$.

In addition to k^{max}, the number of iterations that the process performs, our algorithm have another parameter, r^*, corresponding to the reinforcement of the elements of τ regarding to the best solution ever produced by the process, π^*. We can now present into details our implementation of our ant system, step by step.

1) Initial values

Initially, variable r is set to 1. This variable corresponds to the reinforcement of the traces for the solution produced at the current iteration. All the traces are set to r : $\tau_{ij} = r$, $1 \le i, j \le n$.

2a) Generation of a new solution

For generating a new solution π, we choose the elements of π successively, in a random order and with a probability proportional to the values contained in the matrix of traces. More formally, the process is the following :

1) $I = \emptyset$, $J = \emptyset$
2) *While* $|I| < n$ *repeat :*
   - 2a) *Choose* i*, randomly, uniformly,* $1 \le i \le n$, $i \notin I$.
   - 2b) *Choose* j*, randomly,* $1 \le j \le n$, $j \notin J$*, with probability* $\tau_{ij} \Big/ \sum_{1 \le k \le n,\, k \notin J} \tau_{ik}$ *and set* $\pi_i = j$.
   - 2c) $I = I \cup \{i\}$, $J = J \cup \{j\}$

2b) Improvement of a solution

The solution π generated at the previous step is generally not so good ; at the first iteration, π is just a random permutation. So, we apply to π the first steps of a first improving neighbour procedure. More formally, we repeat the following procedure twice, where $\Delta(\pi, i, j)$ is the difference in the objective function value when exchanging the elements π_i and π_j in π :

1) $I = \emptyset$.
2) *While* $|I| < n$ *repeat :*
   - 2a) *Choose* i*, randomly, uniformly,* $1 \le i \le n$, $i \notin I$.
   - 2b) $J = \{i\}$
   - 2c) *While* $|J| < n$ *repeat :*
     - 2c1) *Choose* j*, randomly, uniformly,* $1 \le j \le n$, $j \notin J$.
     - 2c2) *If* $\Delta(\pi, i, j) < 0$*, exchange* π_i *and* π_j *in* π*.*
     - 2c3) $J = J \cup \{j\}$
   - 2d) $I = I \cup \{i\}$.

The evaluation of $\Delta(\pi, i, j)$ can be performed in $O(n)$ using the formula :

$$\Delta(\pi, i, j) = \begin{cases} (a_{ii} - a_{jj})(b_{\pi_j\pi_j} - b_{\pi_i\pi_i}) + (a_{ij} - a_{ji})(b_{\pi_j\pi_i} - b_{\pi_i\pi_j}) + \\ \sum_{k \neq i, j} (a_{ki} - a_{kj})(b_{\pi_k\pi_j} - b_{\pi_k\pi_i}) + (a_{ik} - a_{jk})(b_{\pi_j\pi_k} - b_{\pi_i\pi_k}) \end{cases}$$

Therefore this procedure can be executed in $O(n^3)$; it does not necessarily returns a local optimum, but it is fast and may produce different solutions when starting with the same initial, not locally optimal solution. From now on, we are going to call μ the improved solution.

2c) Updates

If the new solution generated with the trace (before applying the improvement steps) is equal to π^* then set $r = r + 1$ and re-set all the traces to r. This corresponds to cancelling all the informations contained in the matrix of the traces when these informations are not significantly different from π^* (i. e. when the traces $\tau_{i\pi_i^*}$ are so high regarding to the other traces that the solution π^* has a high probability of being generated as initial solution). The effect of incrementing r is to give less weight to solution π^* in the further updates of the traces.

If μ, the improved solution is better than π^*, i. e. $f(\mu) < f(\pi^*)$, then set $\pi^* = \mu$, $r = 1$ and re-set all the traces to r.

Finally, we proceed to the regular update of the traces as follows :

1) For i = *1 to* n *do :*

1a) $\tau_{i\mu_i} = \tau_{i\mu_i} + r$

1b) $\tau_{i\pi_i^*} = \tau_{i\pi_i^*} + r^*$

In our implementation, we used $r^* = 4$.

3. NUMERICAL RESULTS.

Several good heuristic methods have been designed for the QAP. Among the best ones, let us quote : A robust taboo search (RTS) due to Taillard (1991), the reactive taboo search of Battiti and Tecchioli (1994) that is especially good for long runs on uniform problem instances, the genetic hybrid of Fleurent and Ferland (1994) that is especially good for long runs on structured problems and finally the star-shaped diversified taboo search of Sondergeld and Voß, that was successfully applied to a turbine runner balancing problem. Very recently, another ant system due to Stützle (1997) has been developed. However, it seems that the computational results of our method are generally better. It is shown in Taillard (1995) that the quality of the solutions produced by a given method highly depends on the type of the problem instance. Our purpose is to show that our ant procedure is able to find good solutions in a moderate amount of time for problems presenting a structure. For this type of problems, the best methods seems to be the genetic hybrid of Fleurent and Ferland followed by RTS. Unfortunately, the genetic hybrid method requires a large amount of computing time. In fact, short runs of this method simply consist in repeating independent RTS with $4n$ iterations. Therefore, we compare our ant system with RTS and with repetitions of short RTS that start with different initial solutions.

Table 1 provides the following informations :

- — *The problem name, as given in the QAPLIB of Burkard, Karisch and Rendl (1991–), the numbers embedded in these names correspond to the size of the problems,*
- — *The best known solution value of these instances,*
- — *The quality of the solutions produced by the ant system after 50 iterations (measured in per cent above the best known value),*
- — *The computing time of the ant system (seconds on Sun Sparc 5),*
- — *The quality of the solutions produced by RTS after 40*n *iterations*
- — *The quality of the best of 10 independent runs of the robust taboo search with 4n iterations,*
- — *The computing time of RTS (which is about the same for both RTS versions).*

Problem name	Best known solution	Ant Quality $k^{max} = 50$ [%]	Ant CPU [s]	RTS 40*n* [%]	10 RTS 4*n* [%]	CPU RTS [s]
Nug20	2570	*0.47*	1.1	*0.26*	*0.35*	1.0
Nug30	6124	1.24	4.4	0.38	0.48	4.2
Sko42	15812	1.22	13	*0.39*	0.58	13
Sko56	34458	1.23	34	0.51	0.75	33
Sko81	90998	1.01	103	0.59	0.74	110
Tai20a	703482	1.18	1.3	1.18	*1.07*	1.3
Tai50a	4941410	3.71	24	1.78	2.06	26
Tai100a	21125314	3.06	220	1.23	1.47	243
Bur26 a-h	5609661.875	*0.027*	1.95	*0.28*	*0.039*	2.4
Els19	17212548	*1.76*	0.85	*26.6*	*2.17*	1.0
Esc64	116	*0*	1.9	*1.55*	*0*	23
Esc128	64	*0*	80.3	13.1	*1.25*	405
Kra30 a-b	90160	*1.86*	4.3	2.02	1.59	4.0
Tai20b	122455319	*0.51*	1.14	*20.8*	*1.35*	1.2
Tai25b	344355646	0.34	2.5	15.1	*1.67*	2.7
Tai30b	637117113	0.93	4.4	13.8	2.72	4.8
Tai35b	283315445	0.71	7.3	10.2	2.25	8.2
Tai40b	637250948	0.86	11	10.3	5.14	12
Tai50b	458821517	0.79	24	6.67	2.46	27
Tai60b	608215054	0.94	46	7.92	2.63	47
Tai80b	818415043	2.36	106	4.78	3.50	120
Tai100b	1185996137	1.11	212	4.35	2.64	231
Tai150b	498998296	1.76	752	3.25	2.35	855

Table 1 : Comparison of the ant system with RTS

All results in Table 1 are average for 10 runs of each method. When a run find the best known solution, it is stopped ; therefore, the average computing times are sometimes lower than the time needed for completing the iteration limit. The fact that at least one of the ten runs have found the best known solution is indicated in italics in the table. We have divided Table 1 into two parts : first are the problems that are unstructured and regular (i. e. the matrices are either randomly, uniformly generated or corresponds to manhattan distances on a grid) ; the second part corresponds to problems having a strong, non uniform structure. We see in this table that our ant system performs very well on that last type of problems while being weaker on unstructured problems. Let us quote that longer runs of our ant system have succeeded in improving the best known solutions to the largest problems of the QAPLIB : tai150b (498998296) and tai256c (44765184).

4. CONCLUSIONS

We have proposed a new heuristic for solving the quadratic assignment problem. This approach is based on the ant system, but includes few non standard variants : each ant uses a local search procedure, the traces are updated in a very simple way and are frequently cancelled, leading to a method that is rather aggressive.

Numerical experiments show that the method is competitive on structured problems, leading to improving two of the largest instances of the QAPLIB, but performs fairly on random, uniform ones.

Acknowledgements

This research was supported by the Swiss National Science Foundation project number 21-45653.95.

5. BIBLIOGRAPHY

R. Battiti and G. Tecchiolli, “The reactive tabu search”, *ORSA Journal on Computing 6*, 1994, 126-140.

R. E. Burkard, S. Karisch and F. Rendl, “QAPLIB — A quadratic assignment problem library”, *European Journal of Operational Research 55*, 1991, 115-119, electronic update : http://fmtbhpl.tu–graz.ac.at/~karisch/qaplib, (29. 1. 1997).

A. Corloni, M. Dorigo and V. Maniezzo, “Distributed Optimization by Ant Colonies”, *Proceedings of the European Conference on Artificial Life*, Elsevier Publishing, 1991, 134-142.

M. Dorigo and L. M. Gambardella, “Ant colony system : A cooperative Learning Approach to the Traveling Salesman Problem”, *IEEE Transactions on Evolutionary Computing 1*, 1997.

C. Fleurent and J. Ferland, “Genetic hybrids for the quadratic assignment problem”, *DIMACS Series in Mathematics and Theoretical Computer Science 16*, 1994, 190-206.

É. Taillard, “Robust taboo search for the quadratic assignment problem”, *Parallel Computing 17*, 1991, 443-455.

L. Sondergeld and S. Voß, “A Star-Shaped Diversification Approach in Tabu Search”, in : *Meta-Heuristics : Theory and Applications*, I. H. Osman and J. P. Kelly (editors), Kluwer Academic Publishers, 1996, 489-502.

T. Stützle, “The MAX-MIN Ant System and Local Search for Combinatorial Optimization Problems”, Technical report of the Department of Computer Science - Intellectics Group of the Technical University of Darmstadt, 1997.