Information Exchange in Multi Colony Ant Algorithms

Martin Middendorf[1], Frank Reischle[2], Hartmut Schmeck[3]

Institut für Angewandte Informatik
und Formale Beschreibungsverfahren, Universität Karlsruhe,
D-76128 Karlsruhe, Germany
{[1]middendorf,[3]schmeck}@aifb.uni-karlsruhe.de
[2]Frank.Reischle@gmx.net

Abstract. Multi colony ant algorithms are evolutionary optimization heuristics that are well suited for parallel execution. Information exchange between the colonies is an important topic that not only influences the parallel execution time but also the optimization behaviour. In this paper different kinds of information exchange strategies in multi colony ant algorithms are investigated. It is shown that the exchange of only a small amount of information can be advantageous not only for a short running time but also to obtain solutions of high quality. This allows the colonies to profit from the good solutions found by other colonies and also to search in different regions of the search space by using different pheromone matrices.

1 Introduction

Ant Colony Optimization (ACO) is a metaheuristic to solve combinatorial optimization problems by using principles of communicative behaviour occurring in ant colonies. Ants can communicate information about the paths they found to food sources by marking these paths with pheromone. The pheromone trails can lead other ants to the food sources.

ACO was introduced by Dorigo et al. [4, 6]. It is an evolutionary approach where several generations of artificial ants search for good solutions. Every ant of a generation builds up a solution step by step thereby going through several decisions until a solution is found. Ants that found a good solution mark their paths through the decision space by putting some amount of pheromone on the edges of the path. The following ants of the next generation are attracted by the pheromone so that they will search in the solution space near good solutions.

Ant algorithms are good candidates for parallelization but not much research has been done in parallel ant algorithms so far. One reason might be that it seems quite obvious how to parallelize them. Every processor can hold a colony of ants and after every generation the colonies exchange information about their solutions. Then, every colony computes the new pheromone information which is usually stored in some pheromone matrix. Most parallel implementations of ant algorithms follow this approach and differ only in granularity and whether the computations for the new pheromone matrix are done locally in all colonies or centrally by a master processor which distributes the new matrix to the colonies. Exceptions are [3, 9, 10] where the colonies exchange information only after several generations (Similar principles are also used in the island model of genetic algorithms, e.g. [8]). The effect is that the pheromone matrices of the colonies may now differ from each other.

In this paper we investigate different kinds of information exchange between colonies of ants. We show that it can be advantageous for the colonies to exchange not too much information not too often so that their pheromone matrices can develop independently to some extent. This allows the colonies to exploit different regions in the solution space.

2 Ant Algorithm for TSP

The Traveling Salesperson problem (TSP) is to find for n given cities a shortest closed tour that contains every city exactly once. Several ant algorithms have been proposed for the TSP (e.g. [2, 6, 12]). The basic approach of this algorithms is described in the following.

In each generation each of m ants constructs one solution. An ant starts from a random city and iteratively moves to another city until the tour is complete and the ant is back at its starting point. Let d_{ij} be the distance between the cities i and j. The probability that the ant chooses j as the next city after it has arrived at city i where j is in the set S of cities that have not been visited is

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha} [\eta_{ij}]^{\beta}}{\sum_{k \in S} [\tau_{ik}]^{\alpha} [\eta_{ik}]^{\beta}}$$

Here τ_{ij} is the amount fo pheromone on the edge (ij), $\eta_{ij} = 1/d_{ij}$ is a heuristic value, and α and β are constants that determine the relative influence of the pheromone values and the heuristic values on the decision of the ant.

For some constant $m_{best} \leq m$ the m_{best} ants having found the best solutions are allowed to update the pheromone matrix. But before that is done some of the old pheromone is evaporated by multiplying it with a factor $\rho < 1$

$$\tau_{ij} = \rho \cdot \tau_{ij}$$

The reason for this is that old pheromone should not have a too strong influence on the future. Then every ant that is allowed to update adds pheromone to every edge (ij) which is on its tour. The amount of pheromone added to such an edge (ij) is Q/L where L is the length of the tour that was found and Q is a constant:

$$\tau_{ij} = \tau_{ij} + \frac{Q}{L}$$

To prevent that too much of the pheromone along the edges of the best found solution evaporates an elitist strategy is used. In every generation e additional ants - called elitist ants - add pheromone along the edges of the best known solution.

The algorithm stops when some stopping criterion is met, e.g. a certain number of generations has been done or the best found solution has not changed for several generations.

3 Parallel Ant Algorithms

Only a few parallel implementations of ant algorithms have been described in the literature. A very fine-grained parallelization where every processor holds only a single ant was implemented by Bolondi and Bondaza [1]. Due to the high overhead for communication this implementation did not scale very well with a growing number of processors. Better results have been obtained by [1, 5] with a more coarse grained variant.

Bullnheimer et al. [3] propose a parallelization where an information exchange between several colonies of ants is done every k generations for some fixed k. They show by using simulations how much the running time of the algorithm decreases with an increasing interval between the information exchange. But it is not discussed how this influences the quality of the solutions.

Stützle [11] compares the solution quality obtained by the execution of several independent short runs of an ant algorithm with the solution quality of the execution of one long run whose running time equals the sum of the running times of the short runs. Under some conditions the short runs prooved to give better results. Also, they have the advantage to be easily run in parallel and it is possible to use different sets of parameters for the runs.

Talbi et al. [13] implemented a parallel ant algorithm for the Quadratic Assignment problem. They used a fine-grained master-worker approach, were every worker holds a single ant that produces one solution. Every worker then sends its solution to the master. The master computes the new pheromone matrix and sends it to the workers.

An island model approach that uses ideas from Genetic Algorithms was proposed in Michels et al. [10]. Here, every processor holds a colony of ants exchanging the locally best solution after every fixed number of iterations. When a colony receives a solution that is better than the best solution found so far by this colony, the received solution becomes the new best found solution. It influences the colony because during trail update some pheromone is always put on the trail that corresponds to the best found solution.

The results of Krüger et al. [9] indicate that it is better to exchange only the best solutions found so far than to exchange whole pheromone matrices and and add the received matrices — multiplied by some small factor — to the local pheromone matrix.

4 Strategies for Information Exchange

We investigate four strategies for information exchange differing in the degree of coupling that is enforced between the colonies through this exchange. Since the results of [9] indicate that exchange of complete pheromone matrices is not advantageous all our methods are based on the exchange of single solutions.

1. Exchange of globally best solution: In every information exchange step the globally best solution is computed and sent to all colonies where it becomes the new locally best solution.
2. Circular exchange of locally best solutions: A virtual neighbourhood is established between the colonies so that they form a directed ring. In every information exchange step every colony sends its locally best solution to its successor colony in the ring. The variable that stores the best found solution is updated accordingly.
3. Circular exchange of migrants: As in (2) the processors form a virtual directed ring. In an information exchange step every colony compares its m_{best} best ants with the m_{best} best ants of its successor colony in the ring. The m_{best} best of these $2m_{best}$ ants are allowed to update the pheromone matrix.
4. Circular exchange of locally best solutions plus migrants: Combination of (2) and (3).

5 Results

We tested our information exchange strategies on the TSP instance eil101 with 101 cities from the TSPLIB [14]. The smallest tourlength for this instance is known to be 629. The parameter values that were used are: $\alpha = 1$, $\beta = 5$, $\rho = 0.95$, $Q = 100$, $e = 10$. All runs were done with $m = 100$ ants that are split into $N \in \{1, 5, 10, 20\}$ colonies. The number m_{best} of ants that are allowed to update the pheromone matrix in a colony was varied with the size of the colony. For colonies of size 100 three ants update, for colonies of size 10 or 20 two ants update, and for colonies of size 5 ants only one ant updates. For strategies (3) and (4) m_{best} migrants where used. Every run was stopped after 500 generations. All given results are averaged over 20 runs.

Some test runs were performed to show the influence of the information exchange strategy on the differences between the pheromone matrices of the colonies. Let $T^{(k)}$ be the pheromone matrix of the kth colony. Then, the average pheromone matrix M is defined by

$$M_{ij} = \frac{1}{N} \sum_{k=1}^{N} T_{ij}^{(k)}$$

To measure the average difference between the pheromone matrices we took the average σ of the variances between the single elements of the matrices, i.e.,

$$\sigma = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij} \quad \text{where} \quad S_{ij} = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}(T_{ij}^{(k)} - M_{ij}^{(k)})}$$

The test runs were done with $N = 10$ colonies of 10 ants each. Figure 1 shows the results when an information exchange is done after every $I = 10$, respectively $I = 50$ generations. The figure shows that during the first generations the pheromone matrices in the colonies evolve into different directions which results in a growing average difference σ. One extreme is method (1) which grows up to a maximum at $\sigma = 0.9$ (0.23) after 10 generations for $I = 10$ (respectively 50 generations for $I = 50$). An exchange of the global best solution has the effect that the average difference becomes smaller very fast before it starts to grow slowly until the next information exchange occurs. After about 125 (275) generations σ is smaller than 0.01 in the case $I = 10$ (respectively $I = 50$). The other extreme is method (3) where the exchange of migrants has only a small effect on σ. It increases up to about 0.25 and then degrades very slowly to 0.24 in generation 500 for $I = 10$. The curve for $I = 50$ is nearly the same with only slightly larger σ. The curve of method (2) lies between the curves of methods (1) and (3) but there is a big difference between the cases $I = 10$ and $I = 50$. For $I = 50$ the difference σ goes up to 0.24 and then drops slowly down to 0.12 in generation 500. For $I = 10$ the difference σ goes only up to 0.17 and falls below 0.01 at generation 350. Method (4) behaves similarly to method (2) which means the circular exchange of the locally best solution is the dominating factor.

In order to show how strict the ant algorithm converges to one solution the average number of alternatives was measured that an ant has for the choice of the next city. We are interested only in cities that have at least some minimal chance to be chosen next. For the kth ant that is placed on city i let $D^{(k)}(i) = |\{j \mid p_{ij} > \lambda, j \in [1:n] \text{ was not visited}\}|$ be the number of possible successors that have a probability $> \lambda$ to be chosen. Then, the average number of alternatives with probability $> \lambda$ during a generation is

$$D = \frac{1}{mn}\sum_{k=1}^{m}\sum_{i=1}^{n} D^{(k)}(i)$$

Clearly, $D \geq 1$ always holds for $\lambda < 1/(n-1)$. Note that a similar measure — the λ-branching factor — was used in [7] to measure dimension of the search space. In contrast to D the λ-branching factor considers all other cities as possible successors not only those cities that have not yet been visited by the ant. Hence, D takes into account only the alternatives that the ants really meet, whereas the λ-branching factor is a more abstract measure and problem dependent.

Figure 2 shows the influence of the information exchange strategies on D for $\lambda = 0.01$ when information exchange is done every $I = 10$, respectively $I = 50$ generations. The figure shows that after every information exchange step the D value becomes larger. But in all cases it falls down below 2 before the 80th generation. After generation 150 the D values for method (1) are always lower than for the other methods. They are below 1.1 after generation 270 for $I = 10$, respectively after generation 290 for $I = 50$. It is interesting that during the first 100-150 generations D falls fastest for method (3) but in generation 500 the D value of 1.08 in case $I = 10$ is the largest. The D values of methods (2) and (4) with circular information exchange of local best solutions are quite similar. They are always larger than those of method (1). Compared to method (3) they are smaller after generation 300 in case $I = 10$ but are always larger in case $I = 50$.

Table 1 shows the lengths of the best found tours after 500 generations with methods (1)-(4) and for the case that no information exchange takes place when $I = 50$. In the case of no information exchange it is better to have one large colony than several smaller ones (see also Figure 3). It was observed that the length of the solution found by one colony does not change any more after generation 250. For methods (1) and (3) there is no advantage to have several colonies over just one colony. It seems that the exchange of only a few migrants in method (1) is so weak that the colonies can not really profit from it. It should be noted that the picture changes when information exchange is done more often. E.g., for $I = 10$ we found that 5 colonies are better than one (best found solution was 638.65 in this case).

Table 1. Different strategies of information exchange: best found solution after 500 generations, $I = 50$

	No information exchange	Exchange of globally best solution	Circular exch. of locally best solutions	Circular exchange of migrants
N=1	640.15	—	—	—
N=5	642.85	640.70	637.10	643.15
N=10	642.85	641.65	637.10	642.75
N=20	648.00	642.90	640.45	645.45

Methods (2) and (4) where local best solutions are exchanged between neighbouring colonies in the ring perform well. Figure 3 shows that the solutions found with method (2) by 5 or 10 colonies are always better than those of one colony after 250 generations, respectively 350 generations, for $I = 50$. In generation 500 the length of the best solution found by 10 colonies is about the same as that found by 5 colonies. Moreover, the curves show that there is still potential for further improvement after 500 generations for the 10 colonies and the 20 colonies. The curves for method (4) are quite similar to those in Figure 3 and are omitted. Table 2 shows the behaviour of method (2) when the information exchange is done more often, i.e., every 10 or 5 generations. For an exchange after every 5 generations the solution quality found in generation 500 is not or only slightly better for the multi colonies compared to the case with one colony. It seems that in this case the information exchange is too much in the sense that the colonies can not evolve into different directions.

Table 2. Circular exchange of locally best solutions

	I=5	I=10	I=50
N=5	642.30	638.90	637.10
N=10	642.90	638.55	637.10
N=20	639.35	638.20	640.45

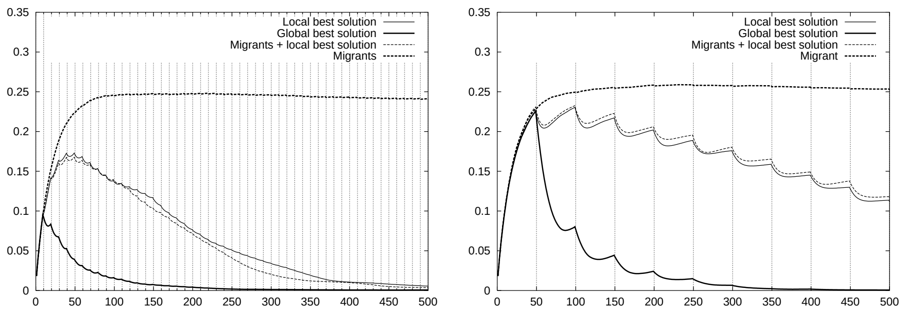

Fig. 1. Difference σ between matrices , Left: migration interval 10, Right: migration interval 50

6 Conclusion

Different methods for information exchange in multi colony ant algorithms were studied. Clearly, ant algorithms with several colonies that exchange not too much information can effectively be parallelized. It was shown that even the solution quality can improve when the colonies exchange not too much information. Instead of exchanging the local best solution very often and between all colonies it is better to exchange the local best solution only with the neighbour in a directed ring and not too often.

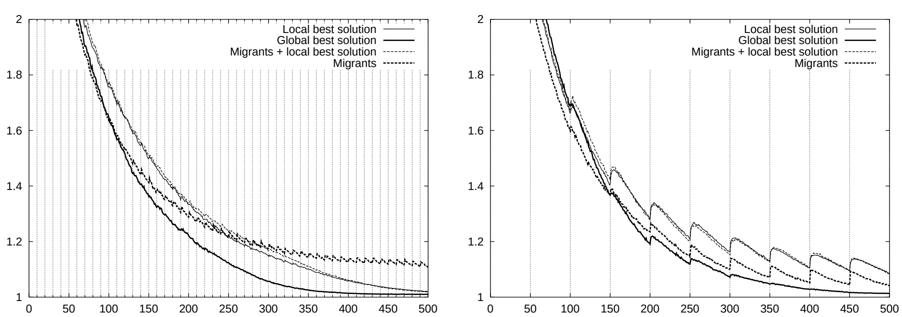

Fig. 2. Average number of alternatives D, Left: migration interval 10, Right: migration interval 50

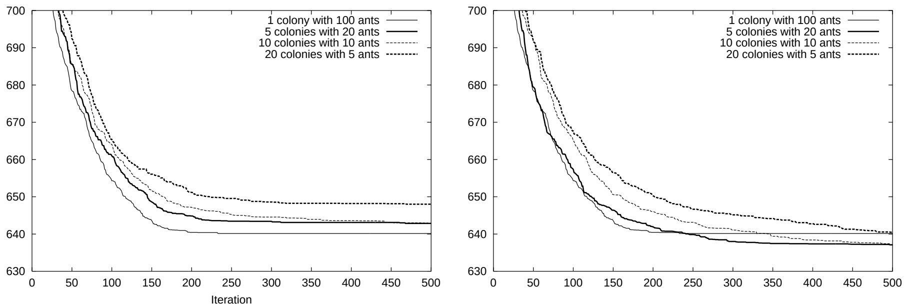

Fig. 3. Best found solution, Left: no information exchange, Right: circular exchange of locally best solution, migration interval 50

References

1. M. Bolondi, M. Bondaza: Parallelizzazione di un algoritmo per la risoluzione del problema del comesso viaggiatore; Master's thesis, Politecnico di Milano, 1993.
2. B. Bullnheimer, R.F. Hartl, C. Strauss: A New Rank Based Version of the Ant System - A Computational Study; CEJOR, Vol 7, 25-38, 1999.
3. B. Bullnheimer, G. Kotsis, C. Strauss: Parallelization Strategies for the Ant System; in: R. De Leone et al. (Eds.), *High Performance Algorithms and Software in Nonlinear Optimization*; series: Applied Optimization, Vol. 24, Kluwer, 87-100, 1998.
4. M. Dorigo: Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, 1992.
5. M. Dorigo: Parallel ant system: An experimental study; Unpub. manuscript, 1993.
6. M. Dorigo, V. Maniezzo, A. Colorni: The Ant System: Optimization by a Colony of Cooperating Agents; *IEEE Trans. Sys., Man, Cybernetics – B*, 26, 29-41, 1996.
7. L. M. Gambardella, M. Dorigo: Ant-Q: A Reinforcement Learning approach to the traveling salesman problem; *Proceedings of ML-95, Twelfth Intern. Conf. on Machine Learning*, Morgan Kaufmann, 252-260, 1995.
8. U. Kohlmorgen, H. Schmeck, K. Haase: Experiences with fine-grained parallel genetic algorithms; *Ann. Oper. Res.*, 90, 203-219, 1999.
9. F. Krüger, M. Middendorf, D. Merkle: Studies on a Parallel Ant System for the BSP Model; Unpub. manuscript, 1998.
10. R. Michels, M. Middendorf: An Ant System for the Shortest Common Supersequence Problem; in: D. Corne, M. Dorigo, F. Glover (Eds.), *New Ideas in Optimization*, McGraw-Hill, 1999, 51–61.
11. T. Stützle: Parallelization strategies for ant colony optimization; in: A. E. Eiben, T. Bäck, M. Schonauer, H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature - PPSN V*, Springer-Verlag, LNCS 1498, 722-731, 1998.
12. T. Stützle, H. Hoos: Improvements on the ant system: Introducing MAX(MIN) ant system; in G. D. Smith et al. (Eds.), *Proc. of the International Conf. on Artificial Neutral Networks and Genetic Algorithms*, Springer-Verlag, 245-249, 1997.
13. E-G. Talbi, O. Roux, C. Fonlupt, D. Robillard: Parallel ant colonies for combinatorial optimization problems; in J. Rolim et al. (Eds.) *Parallel and Distributed Processing, 11 IPPS/SPDP'99 Workshops*, LNCS 1586, Springer, 239-247, 1999.
14. http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB/TSPLIB.html