SIMULATION OF ANT COLONIES WITH HINTS GENERATED BY PARALLEL HEURISTICS

Oleg Kovářík, Miroslav Skrbek

Czech Technical University in Prague, Faculty of Electrical Engineering,
Department of Computer Science and Engineering
Karlovo náměstí 13,
121 35, Prague, Czech Republic

kovarole@fel.cvut.cz, skrbek@fit.cvut.cz(Oleg Kovářík, Miroslav Skrbek)

Abstract

In this paper we present a new approach for combining discrete optimization algorithms, which resulted from a simulation of several scenarios of information exchange between algorithms running in parallel. We show that a group of parallel metaheuristics can create probabilistic “hints” for colony of artificial ants, which solves the same instance of a Traveling Salesman Problem (TSP). These “hints” are realized by creating intersections of solutions from fast independent solvers and they can be used to improve optimization process. A convergence speedup is shown for a group of Simulated Annealing algorithms together with Max-Min Ant System (variant of Ant Colony Optimization) on non-geometric TSP and for a given time window. This effect holds for random instances of sizes in of order of magnitude of 10^3. Presented model of cooperation between algorithms is a first step on the path to a complex metaoptimization system based on combinations of heuristics. We show, that the cooperation of diverse optimization algorithms can in some sense give better results than a homogeneous set of algorithms.

Keywords: Ant Colony Optimization, Traveling Salesman Problem, parallel metaheuristics, algorithm combination

Presenting Author’s Biography

Oleg Kovářík is currently a Ph.D. student at the Faculty of Electrical Engineering, Czech Technical University in Prague. His research interest is a simulation of artificial ant colonies for solving optimization problems. The topic of his concern are mainly improvements of existing algorithms inspired by behavior of ants, that are used for combinatorial and continuous optimization. Last results lead to application of metalearning principles in optimization which corresponds to the topic of his thesis: “Ant Algorithms in Metalearning and Metaoptimization”.

1 Introduction

In last decades, several models of ant colonies were developed to solve combinatorial optimization problems. These metaheuristics are appealing not only for its simple conceivability. What makes this family of algorithms unique is a representation of the gained knowledge. Like in other metaheuristics, we evaluate series of problem solution candidates and try to create new solutions closer to optimum. In contrast to them, we store the knowledge in the probabilistic form. This is in fact a simplified model of real colonies of ants, where the collective behavior of the whole ant colony is driven by the chemical substance called pheromone, that is stored in the environment by all individuals. The model is represented as a matrix of decimal numbers.

We tried to examine several ways to further improve results of optimization process by generating additional hints for the colony of artificial ants. These hints can be generated by other heuristics running in parallel on multi-core processors, that are included in today's computers. We are concerned about benefits of cooperation of diverse optimization algorithms, mainly the question if such a diverse cooperation could outperform sets of same algorithms running in parallel. Diversity is considered an important requirement for both natural and artificial evolution and also No free lunch theorem in search and optimization [1] suggests that each algorithm has its weakness.

We will test our idea on well known benchmark - Traveling Salesman Problem (exhaustive survey including methods mentioned in this paragraph in [2]), because of its simple formulation and easy visualization. Existing similar approach is Tour Merging [3], where the union of several discovered solutions is used as a input graph for further search for better solution. This way the search space is strongly reduced at the cost that we can miss the global optimum. In contrast to this algorithm, we use stronger intersection of several solutions and we use probabilistic representation to leave small probability of exploration for the whole search space. Another interesting approach for combining several solutions for Traveling Salesman Problem is called Alternating Cycles [4].

The main question of this research is whether we can in some cases benefit from combination of optimization algorithms.

2 Background

2.1 Traveling Salesman Problem (TSP)

TSP [2] is a well known combinatorial optimization task often used for testing new ideas in combinatorial optimization and for algorithm benchmarking. One advantage of TSP is that it can be simply informally described. Given a set of cities we want to find a shortest closed path containing all cities. In our experiments, we use standard version with symmetric distances between cities and also asymmetric and non-geometrical version with randomly generated distances between cities.

Many heuristic approaches to solve TSP were developed. In our experiments, we used a number of them, but we will describe only two of them, which were a part of the successful configuration: Max-Min Ant System and Simulated Annealing.

2.2 Max-Min Ant System (MMAS)

MMAS ([5], [6]) is a heuristic from the family of Ant Colony Optimization (ACO) algorithms. We used it, because MMAS usually produces better results than other ACO algorithms. It is inspired by indirect communication in real ant colonies - laying and following the pheromone trails. When searching for food, ants use a pheromone trail (fig.1) to mark their paths between the food and the nest. Other ants are more likely to follow the trails with greater amount of pheromone and if they find enough food, these trails are reinforced. Random deviations from their paths also make it possible to gradually shorten the length of the path. Pheromone evaporation allows to forget less convenient paths.

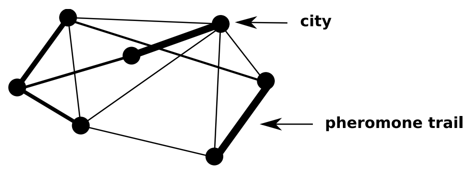

Fig. 1 Schema of TSP problem solved by MMAS. Thicker lines mean higher amount of pheromone and thus higher probability of selection by ant during a path construction.

Important property of this algorithm is that it stores quality of edge between all pairs of cities in the form of matrix of real numbers. This probabilistic representation allows us to adjust the probability of inclusion of the edge in solution. We can use some additional knowledge and simply increase values for selected edges.

MMAS uses ants with a memory containing partial tours between cities. The probability of moving from city i to allowed (unvisited) city j is given by

$$p_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{j \in allowed}[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta} & \text{if } j \in allowed \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\tau_{ij}(t)$ is the amount of pheromone between city i and j, $\eta_{ij} = 1/d_{ij}$ called visibility is computed from distance d_{ij} between cities, and α and β are parameters which allow the user to control the importance of the pheromone and visibility.

After ants constructed their paths, pheromone between every two cities is evaporated and then reinforced:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + Q/L(t) \tag{2}$$

for all edges (i, j) that are present in the best tour found $T(t)$. Here ρ is evaporation factor, and the amount of the pheromone to be added is calculated from the length of the best tour $L^k(t)$ and a constant Q.

MMAS limits the possible amounts of pheromone to the interval $< \tau_{min}, \tau_{max} >$ to keep the probability of choosing each path greater than zero but not extremly high. This helps the algorithm to keep chances of escaping from local optima. Initial amounts are set to τ_{max} to boost up random search in the early iterations. Important technique for improving time complexity is restriction of set of examined cities to a limited neighborhood of the actual city.

2.3 Simulated Annealing (SA)

SA [7] is another probabilistic metaheuristic inspired by metallurgical process of heating and controlled cooling of material. Result of this process is a structure without defects, because atoms form crystals with optimal energetic state. In optimization we repetitively make random changes to the actual solution and with some probability we accept solutions with the worse quality. This probability is slowly decreasing and is a function of parameter called temperature.

Both metaheuristics use random changes, which were in this case implemented by running random pair of distortion procedures from: swapping of two cities, inverting a subpath, shifting a subpath.

2.4 3-OPT

The previous algorithms are useful for global optimization, but to achieve good results, we have to accompany their usage with local search algorithms which are able to lead the optimization process to local optima. Simple local search for TSP is 3-OPT algorithm that consists of repetitive removal of 3 edges from existing solution and consecutive reconnection of partial paths in the best possible way. In [8] authors also describe several important techniques for algorithm speed-up (limited fixed radius search, don't lookup bits, etc.), which are important to implement to achieve reasonable performance for larger problems.

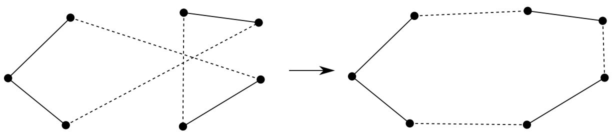

Fig. 2 Example of 3-OPT move. Dotted edges from the left solution are reconnected in the best possible way in the solution on the right side.

3 Proposed models of algorithm combination

Examined models of parallel combination can be divided according to used algorithms:

- homogeneous combination - same algorithms
- heterogeneous combination - different algorithms

and according to the type of information exchange:

0. no exchange - referential
1. the best exchange - immediately distribute new best solution
2. individual pools - maintain pool of solutions (e.g. in Genetic Algorithm), algorithms can ask for individual (solution) from the pool
3. probabilistic pools - maintain probabilistic model of solution (e.g. in Max-Min Ant System), generate new solution on request
4. probabilistic hints - maintain probabilistic model of solution, gain knowledge from other running algorithms and use it to change probabilities

In our experiments, we were untill now successful only with variant 4, which is the subject of this paper. For other variants we were not able to reach any type of significant improvement in comparison with variant 0 - parallel independent execution of several instances of the best heuristic from the implemented set. Algorithm with probabilistic hints will be described in the next section.

3.1 Ant Colonies with Hints Generated by Parallel Heuristics

Our approach can be divided into two parallel subsystems. The first of them is creation of set of edges which are very likely to appear in good solutions. We construct this set by calculating intersection of several solutions. Edges that appear in all (different) solutions are stored as a "hint" for the second subsystem. Example of such intersection is in fig. 3. In our experiments, we used Simulated Annealing heuristics (with 3-OPT) running in parallel and we calculated the intersection regularly.

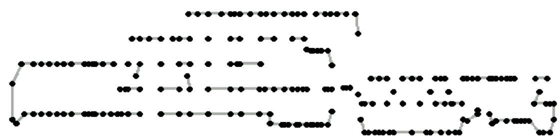

Fig. 3 Part of TSP problem with marked intersection of six solutions from independent Simulated Annealing runs. This information can be used as a hint for Max-Min Ant System algorithm to speed up its convergence.

The second component is Max-Min Ant System algorithm (with 3-OPT) with matrix of probabilities of transitions from one city to the next one. Again, this algorithm is running in parallel with Simulated Annealing algorithms from the first subsystem. In regular intervals, hints are stored into the pheromone matrix and in this way probability of including promising edges in the new solutions is increased. Diversity of solutions is decreasing with time which means that number of edges present in all solutions is increasing (see fig. 4).

To update pheromone on appropriate edges we have to modify equation 2 for edges from the set of hints:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sigma \cdot Q/L(t) \qquad (3)$$

where σ is reinforcement factor for hints, that can be set for example to 2 to achieve twice as high probability for those edges. For edges included in the so far best tour but missing in the intersection of SA results, the original equation holds ($\sigma = 1$).

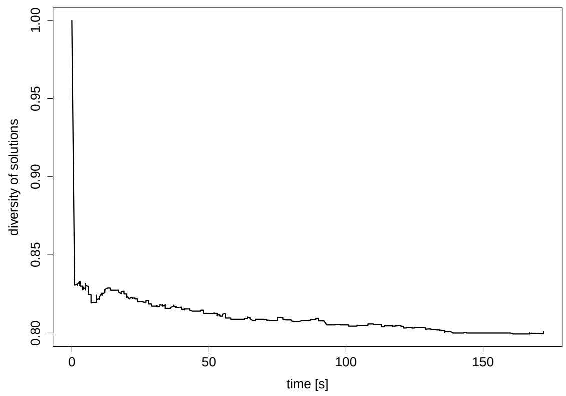

Fig. 4 Diversity of solutions from three Simulated Annealing algorithms is decreasing in time. Value on y axis shows ratio of edges that are not present in solutions from all three SAs.

4 Experiments

For our experiments we used TSP data generator used in 8th DIMACS competition and described in [9]. Instances of size $2-5\times10^3$ were used for all three types: E - uniform, C - clustered and M - non-geometric. Notation E3k.0 means uniformly distributed 3000 cities with generator random seed set to 3000 + 0. We tested all strategies of algorithm cooperation on them, comparing it with results of independent algorithms run. Sizes were chosen to fit in memory (note that MMAS requires n^2 space to maintain the whole pheromone matrix), running time was reasonably long to cover important changes in performance of different algorithms. We used four core processor and ran four algorithms in separate threads. All experiments were repeated 30 times.

In presented graphs, we use the following shortcuts:

- SA4 - four parallel SA algorithms without exchange of solutions
- MMAS4 - four parallel MMAS algorithms without exchange of solutions
- MMAS1,SA3 - one MMAS and three SA algorithms without exchange of solutions
- MMAS1+SA3hints - one MMAS with hints from three SA algorithms

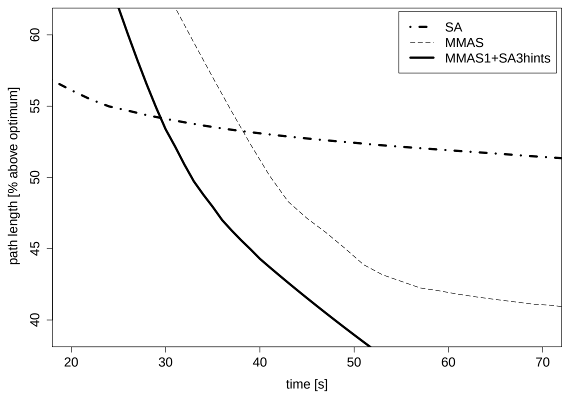

Fig. 5 Comparison of average results for SA algorithms, MMAS algorithms and MMAS updated by intersection of three SA solutions (solid line) for dataset M3k.0. Our MMAS algorithm with hints shows faster convergence in specific time window.

5 Results

For uniform (E) and clustered (C) instances solutions move around 1% above optimal length and our method does not perform better than four parallel independent Simulated Annealing algorithms. However, for random distance matrices (M) the situation differs. Local search is not able to reach near-optimal values quickly and so there is space left for improvements, because of the gap in tens of percent between optimal value and solutions found in few minutes on today's computers. Typical development of solution quality can be seen in fig. 5. MMAS algorithm starts more slowly than Simulated Annealing counterparts, but it begins to converge faster after the initial stage. Speed of convergence is even faster when using hints to increase pheromone levels. At some point solutions generated by MMAS algorithm become shorter than those generated by SA.

The same behavior was observed for instances M2k.0, M3k.0, M4k.0 and M5k.0. Average results for M3k.0 instance show (see fig. 6) how our algorithm outperforms other combinations in time window from 30 to 50 seconds. After this phase, the faster convergence is counterbalanced by stagnation and variants of algorithms with no exchange of solutions become better.

Out algorithm with hints performs significantly the best at around 40 seconds (see fig. 7) for M3k.0 problem. Similar results were observed for other M instances in different time moments. On larger instances, the observed effect appeared later and lasts longer which was expected.

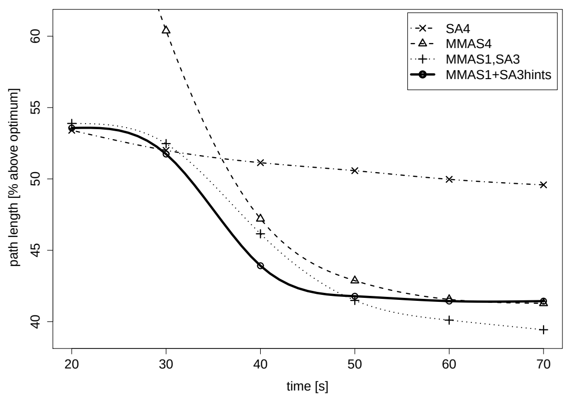

Fig. 6 Comparison of average results from the best four combinations of algorithms on M3k.0 non-geometrical random instance. In time window 30 to 50 seconds, our MMAS algorithm with hints from intersection of SA algorithms outperforms other approaches.

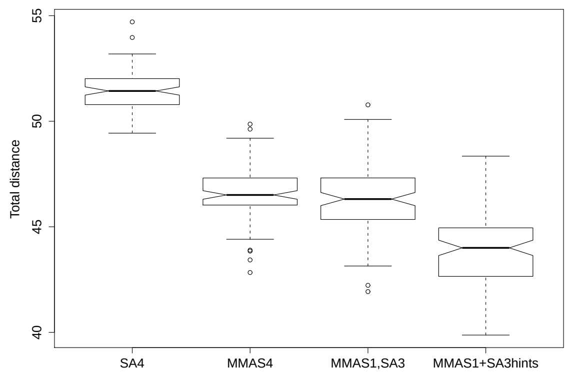

Fig. 7 Results for M3k.0 non-geometrical instance of TSP with 3000 cities 40 seconds after start. Similar results were obtained for instances of size 2-5 thousands

6 Conclusion

We proposed a model of artificial ant algorithm supported by hints from metaheuristics running in parallel. This extended algorithm showed significantly faster convergence speed for several non-geometrical instances of Traveling Salesman Problem. We observed this behavior only in particular time window of algorithm run, however, we can use this result as an encouragement for further research. We achieved faster convergence at the cost of ending in local optimum sooner than in the case of non-cooperating algorithms.

Our result shows, that there exist such conditions for which algorithm combination is beneficial. It implies that we may be able to systematically collect such pieces of knowledge and use them to choose better combinations of algorithms or their parameters to fit selected instances of data and user specified time limitations. This approach that is an example of metalearning (“learning about learning“) will be subject of our future research.

The proposed methods can be further extended by Tour Merging approach [3] to further reduce search space by examining only edges which appeared in union of solutions from all running algorithms, while generating hints from intersection of these solutions. This would be suitable for a very large instances and would be also a subject of our future work.

7 Acknowledgements

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS10/199/OHK3/2T/13 and the research program Transdisciplinary Research in the Area of Biomedical Engineering II (MSM6840770012) sponsored by the Ministry of Education, Youth and Sports of the Czech Republic.

8 References

[1] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[2] David L. Applegate, Robert E. Bixby, Vasek Chvatal, and William J. Cook. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, Princeton, NJ, USA, 2007.

[3] William Cook and Paul D. Seymour. Tour merging via branch-decomposition. *INFORMS Journal on Computing*, 15(3):233–248, 2003.

[4] Hisao Tamaki. Alternating cycles contribution: a strategy of tour-merging for the traveling salesman problem. Research Report MPI-I-2003-1-007, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, March 2003.

[5] Thomas Stützle and Holger Hoos. Max-min ant system and local search for the traveling salesman problem. In *IEEE International Conference on Evolutionary Computation (ICEC ’97)*, pages 309–314. IEEE Press, 1997.

[6] Thomas Stützle and Holger Hoos. Max-min ant system and local search for the traveling salesman problem. In *IEEE International Conference on Evolutionary Computation*, pages 309–314, 1997.

[7] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[8] David S. Johnson and Lyle A. McGeoch. The traveling salesman problem: A case study in local optimization. In *Local Search in Combinatorial Optimization, E.H.L. Aarts and J.K. Lenstra (Eds*, pages 215–310. John Wiley and Sons, Ltd, 1997.

[9] David S. Johnson and Lyle A. McGeoch. Experimental analysis of heuristics for the stsp. In *Local Search in Combinatorial Optimization*. Wiley & Sons, 2001.