Parallel Multicolony ACO Algorithm with Exchange of Solutions

Max Manfrin Mauro Birattari Thomas Stützle Marco Dorigo

IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium

The availability of parallel architectures at low cost, e.g. clusters of PCs connected through fast local networks like Gigabit Ethernet, has widened the interest for the parallelization of algorithms [1]. There are two reasons for parallelizing a metaheuristic if one is interested in performance: (i) given a fixed time to search, the aim is to increase the quality of the solutions found in that time; (ii) given a fixed solution quality, the aim is to reduce the time needed to find a solution not worse than that quality.

We use the Traveling Salesman Problem (TSP), an $\mathcal{NP}$-hard problem, as a case study for testing the impact on the final solution quality reached, given a fixed run time, of the exchange of solutions among multiple colonies on different interconnection topologies. To solve the TSP we use $\mathcal{MAX}-\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) [5], currently one of the best-performing ant colony optimization (ACO) algorithms [3]. Our implementation of $\mathcal{MM}$AS is based on the publicly available ACOTSP code (http://www.aco-metaheuristic.org/aco-code/). To have a version that is easily parallelizable, we removed the occasional pheromone re-initialization applied in the $\mathcal{MM}$AS, and we use only a best-so-far pheromone update. Our version uses the 3-opt local search and quadrant nearest neighbor lists.

The topologies we consider are: fully-connected, replace-worst, hypercube, and unidirectional ring. For each topology we have developed two versions of the algorithm: a first one in which the communication is synchronous, and a second one in which the communication is asynchronous. We consider also the parallel independent runs (PIR) model in which k copies of the same sequential $\mathcal{MM}$AS algorithm are simultaneously and independently executed using different random seeds. The final result is the best solution among all the k runs. These topologies allow us to consider decreasing communication volumes, moving from more global communication, as in fully-connected, to more local communication, as in ring, to basically no communication, as in PIR.

The communication strategy we adopt involves the exchange of best-so-far solutions every r iterations, after an initial period of "solitary" search. A colony *injects* in his current solution-pool a received best-so-far solution if and only if it is better than its current best-so-far solution, otherwise it disregards it. The main advantage of using best-so-far solutions over pheromone matrices is that less data has to be exchanged: for the smallest instance that we consider, each pheromone matrix requires several megabytes of memory space, while a solution requires only some kilobytes.

All algorithms are coded in C using LAM/MPI 7.1.1 communication libraries. Experiments were performed on a homogeneous cluster of 4 computational nodes running GNU/Linux Debian 3.0 as Operating System. Each computational node contains two AMD Opteron 244 CPUs, 2 GB of RAM, and one 1 GB Ethernet network card. The nodes are interconnected through a 48-ports Gbit switch.

Computational experiments are performed with 8 colonies of 25 ants each that exchange the best-so-far solution every 25 iterations, except for the first 100 iterations. We considered 10 instances from TSPLIB with a termination criterion based on run time, dependent on the size of the instance. For each of the 10 instances, 10 runs were performed. In order to have a reference algorithm for comparison, we also test the equivalent sequential $\mathcal{MM}$AS algorithm. We considered two cases: in the first one (SEQ), it runs for the same overall wall-clock time as a parallel algorithm (8-times the wall-clock time of a parallel algorithm), while in the second one (SEQ2), it runs for the same wall-clock time as one CPU of the parallel algorithm.

To compare and aggregate results across different instances, we normalize the results with respect to the distance from the known optimal value. We refer the reader interested in the raw data to the URL: http://iridia.ulb.ac.be/supp/IridiaSupp2006-001/ and to [4].

Our hypothesis is that the exchange of best-so-far solutions among different colonies speeds up the search for high quality solutions, having a positive impact on the performance of the algorithms. Our experimental setup allows us to use statistical techniques for verifying if differences in solutions quality found by the algorithms are statistically significant.

The computational results indicate that all the parallel models perform on average better than the equivalent sequential algorithm, but that the best performing approach is PIR. The differences in performance of all the parallel models with information exchange from those of PIR are statistically significant w.r.t. the *Wilcoxon rank sum* test [2] with *p-values* adjusted by Holm's method, while differences in performance among interconnection topologies are not statistically significant.

The impact of communication on performance seems, therefore, negative. One reason might be that the run times are rather high, and $\mathcal{MM}$AS easily converges in those times.

The modification we implemented to have a version of $\mathcal{MM}$AS that is easily parallelizable result in a stagnation behavior of the sequential algorithm; this stagnation behavior can be avoided to a large extent by parallel independent runs, which also explains its overall good behavior, biasing the performance in favor of PIR over all the other parallel models. An apparent problem of our communication scheme is that communication is too frequent. To better understand the impact that the frequency of communication has on performance, we change the communication scheme to an exchange every $n/4$ iterations, except during the first $n/2$, where n is the size of the instance. The computational results of the new communication scheme on the parallel models replace-worst and ring show that the reduced frequency in communication has indeed a positive impact on the performance of the two parallel algorithms, even though this is not sufficient to achieve better performance w.r.t. PIR.

We believe that better performance than PIR can be obtained by the parallel models either adding the restarting feature, or implementing communication schemes that avoid early convergence. This second approach could be achieved implementing the acceptance of solutions from other colonies only when they "differ" less than a certain number of components, leading to the creation of groups of colonies that search in different areas of the search space, or by exchanging the solutions with a frequency that depends on both, instance size and run time.

Acknowledgments

This work is supported by 'COMP^{2}SYS' and by the 'ANTS' project. 'COMP^{2}SYS' is a Marie Curie Early Stage Training Site, funded by the European Community's Sixth Framework Programme under contract number MEST-CT-2004-505079. The 'ANTS' project is an Action de Recherche Concerte funded by the Scientific Research Directorate of the French Community of Belgium. M. Dorigo and T. Stützle acknowledge support from the Belgian National Fund for Scientific Research (FNRS), of which they are a Research Director and a Research Associate, respectively.

References

[1] Enrique Alba, editor. *Parallel Metaheuristics: A New Class of Algorithms*. Wiley Series on Parallel and Distributed Computing. Wiley-Interscience, Hoboken, NJ, 2005.

[2] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, New York, NY, third edition, 1999.

[3] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.

[4] Max Manfrin, Mauro Birattari, Thomas Stützle, and Marco Dorigo. Parallel ant colony optimization for the traveling salesman problem. In M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, editors, *Ant Colony Optimization and Swarm Intelligence, 5th International Workshop, ANTS 2006*, volume 4150 of *Lecture Notes in Computer Sciences*, Berlin, Germany, Sep 2006. Springer-Verlag.

[5] T. Stützle and H. H. Hoos. $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. *Future Generation Computer System*, 16(8):889–914, Jun 2000.