Presenter: Max MANFRIN
Affiliation: IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
Position: Early-stage researcher at IRIDIA and PhD student in Sciences Appliques at ULB
Title: Parallel Ant Colony Optimization for the Traveling Salesman Problem

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. 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), currently one of the best-performing ant colony optimization (ACO) algorithms. 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 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. We investigate 2 main factors: interconnection topologies and communication modes. The interconnection topologies we consider are: fully-connected, hypercube, ring, and isolated. The communication modes we condiser are: synchronous, asynchronous, and no-communication. Considering these factors we implemented a total of 11 algorithms. 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. We refer the reader interested in the raw data to the URL: http://iridia.ulb.ac.be/supp/IridiaSupp2006-001/

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 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. 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 with *p-values* adjusted by Holm's method, while differences in performance among interconnection topologies are not statistically significant. 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.