Gpu-accelerated Neighborhood Operators For Permutation-based Problems

Victor Machado[1], Sávio Dias [1], Igor Coelho [2], Luiz Ochi [1] , Esteban Clua [1]
victormachado@id.uff.br, sdias@ic.uff.br, igor.machado@ime.uerj.br, satoru@ic.uff.br, esteban@ic.uff.br
[1] Fluminense Federal University [2] Rio de Janeiro State University

What is this poster about ?

This poster presents an efficient GPU implementation of four neighborhood operators that are commonly applied in the local search of many metaheuristics for different permutation-based problems.

Extended Abstract

Several classical Combinatorial Optimization(CO) problems have their solutions partially or completely represented as a permutation of the input data such as the Traveling Salesman Problem (TSP) and the Single Row Facility Layout Problem. Using the same solution representation may allow the application of the same neighborhood operators with only few adaptations to these different problems.

The local search procedure of metaheuristics consists in evaluating neighbors of the current solution usually analyzing all possible moves. There are some well-known operators widely applied to this kind of problems. In this work we consider the following moves: swap, or-opt-1, or-opt-2 and 2-opt. All of them can be performed in $O(n^2)$.

Although many optimization problems have been solved through GPU-parallelization in the last few years, the authors are not aware of a thorough analysis of the neighborhood moves. Therefore, we perform an evaluation of the neighborhood operators rather than analyzing a specific metaheuristic.

In order to conduct the analysis of the proposed implementation the TSP was used as basis due to its large use for benchmark purposes in CO area. The parallel approach achieved good results when compared to the CPU version reaching speedups ranging from 14 to 68 times.

Implementation Details

The TSP instances provide the location of the vertices in Euclidean coordinates. In the CPU implementation it is common to read the coordinates and to calculate the distances at the beginning of the algorithm because this data will be accessed several times. Therefore, when evaluating the change in the objective function we can access the distance directly.

The suggested GPU implementation reduces the data transfer between GPU and CPU by using a device function to calculate the distances through the coordinates. When integrated in a metaheuristic the only memory transfer needed at each iteration is the current solution.

The improvement provided by each possible move is found in the GPU using a two-dimensional indexation for the grid and each thread evaluates a single move. A reduction is performed in each block to find the maximum value of the improvements and its indices.

Finally, a custom atomic function is used to return the global results. The amount of shared memory used does not depends on the input data size, thus there is no limit for the instance size differently from other implementations.

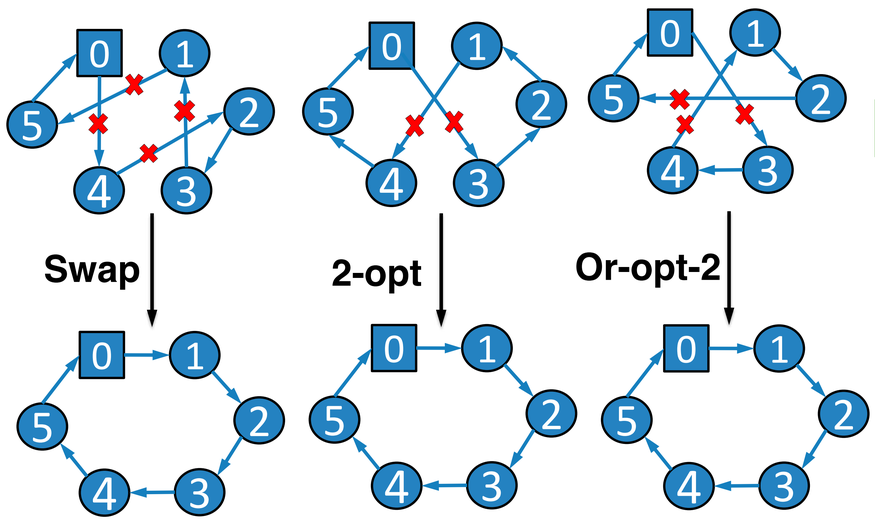

Figure 1: Illustration of the neighborhood moves.

Computational Results

The implementation was tested on the TSPLIB data, that provides several benchmark instances for the TSP: The size of the instances tested varies from 100 to over 18000 vertices. Our experimental platform is composed by a CPU Intel i7 3.60 GHz (only one CPU core was used) and a Nvidia Tesla K20 GPU with 2496 cores. The proposed algorithm was implemented in CUDA version 7.0.

The results obtained are summarized in Table 1 and show the minimum, average and maximum speedups for each neighborhood operator. As shown, substantially high speedups were obtained. Besides that, the proportion of the kernel time was always superior to 90 % in comparison to the total time for all instances greater than 100 vertices, pointing out the reduced use of data transfer between the GPU and the CPU.

Operator	Minimum	Average	Maximum
1-or-opt	15.45	39.09	68.38
2-or-opt	19.67	38.70	65.70
Swap	14.37	26.98	45.85
2-opt	15.75	30.07	51.65

Table 1: Speedups of our GPU implementation

Acknowledgements

This project was financially supported by CAPES-Brazil and CNPQ-Brazil.

References

Schulz, Christian. "Efficient local search on the GPU-investigations on the vehicle routing problem." *Journal of Parallel and Distributed Computing* (2013): 14-31.

Todosijević, Raca, et al. "A general variable neighborhood search variants for the travelling salesman problem with draft limits." *Optimization Letters* (2014) : 1-10.