Markov Chains Interactions on GPU-based Massive Parallel Simulated Annealing: A case study with the Faculty Assignment Problem

Dionisio Chiuratto Agourakis - Aeronautics Institute of Technology (ITA)/J!Quant - Brazil
Vicente Cuéllar Moro - Wavecrafters Consultants - Spain

wavecrafters

The Problem

The UCT – University Course Timetabling has been well studied by the scientific community in the past 25 years, especially in the Artificial Intelligence and Optimization groups, as it is shown to be NP-Complete [3].

The Faculty Assignment Problem - FAP is a derivation of UCT, where the main goal is to assign the faculty to classes, maximizing their "function of preference" subject to classical hard conflict constraints.

Although this is FAP's classical definition, of complexity P – therefore solvable through classical Operations Research techniques like Branch-and-Bound – this work considers a slightly different problem: the Faculty Assignment Problem in a pre-scheduled timetable of classes.

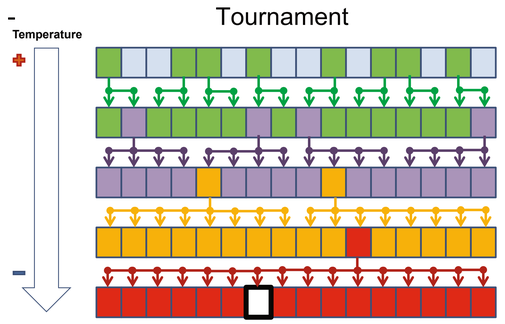
This modification turns the problem into a particular graph-coloring reductible problem, which is well-known for its NP-Completeness, hence assigning to it this same complexity [3].

Massive GPU-based Simulated Annealing

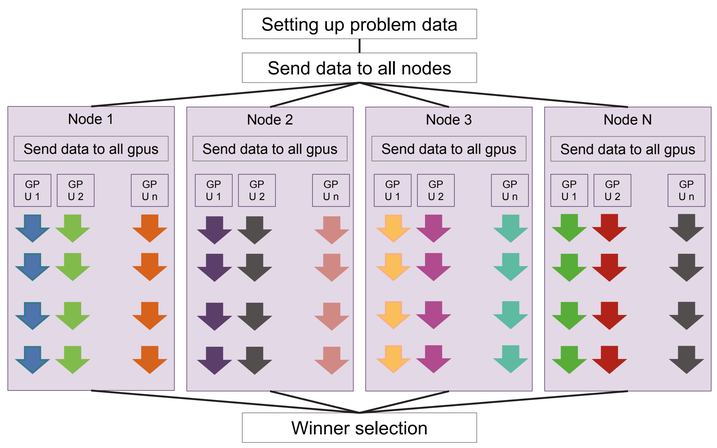
The work uses the Simulated Annealing metaheuristic[1] to attempt the problem, as it is known for being successfully applied to highly constrained problems, while being straightforward to model, differently than other methods such as Genetic Algorithms and Ant Colony Optimization, which requires a great deal of modeling effort to adequate the problems to its scope.

The main challenge about using Simulated Annealing for tackling large-scale instances (>100 professors, >3000 slots) is its long run-time and strictly serial way of work.

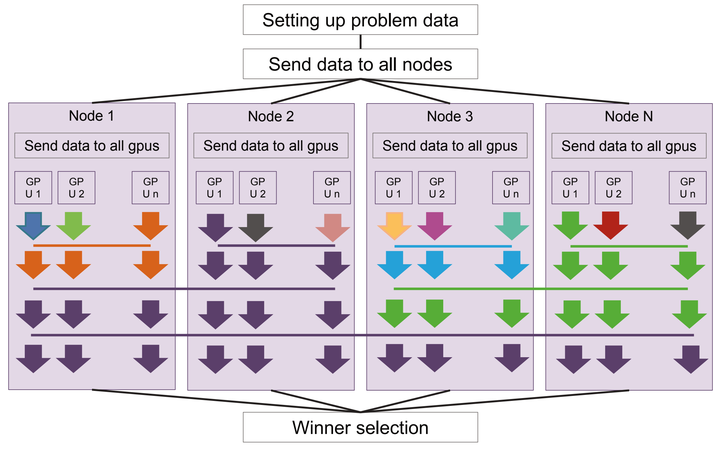
The solution found and proposed here is the massive parallelization of Simulated Annealing algorithm, using NVIDIA CUDA C [2] language to program GPU's (Graphical Processing Units) to do the work.



Multi -Node Multi-Gpu workflow –Markov Chains Mode



Multi -Node Multi-Gpu workflow – Tournament Mode



Tournament Interactions

In parallel simulated annealing, is possible to speedup convergence by adding periodic interactions between the annealing chains, where the best result found so far is broadcasted as the new starting state for the other chains [5]. However, we must make sure the broadcast overhead doesn't eliminate all the benefits achieved, as note [4], and that was possible due to the GPU architecture.

The tournament used in this work was based in []. It works by comparing elements in powers of two, (2 by 2, then 4 by 4, ..., then the whole set of solutions at final iteration). It is important to calibrate carefully the breadth (number of parallel annealing chains) and the depth (number of steps between chain interactions), as it is found to affect both performance and the quality of the solutions.

Configuration and Dataset

The computer used for this experiment was equipped with an intel i7 2600k processor, 16gb RAM and a Geforce GTX 780ti with 3gb RAM.

The dataset was supplied by a well-known brazilian university, and contained 218 professors to be assigned to 3448 pre-scheduled classes of 485 different subjects. There was 1268 different combinations of feasible professor-subject relationships.

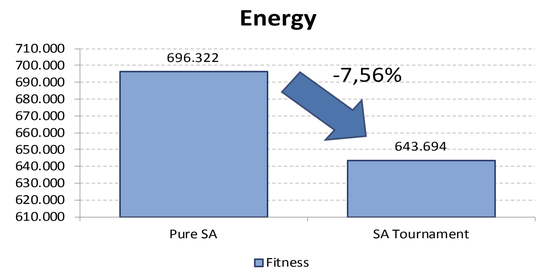
Results

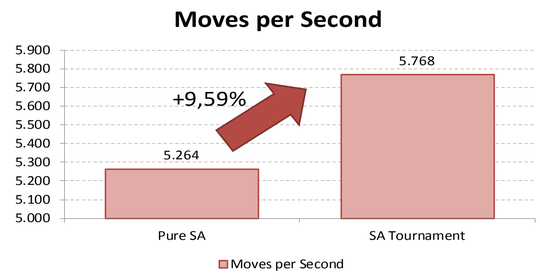
Parameters:

Initial Temp: 70
Final Temp: 1.5
Cooling: Geometric
Factor: 0.8
Iterations/Temp: 10k
Parallel Solutions: 512

EnergyFunction:
Weighted Sum

Weights Used:
Conflicts: 5
Cost: 5
No Professor: 2





The GPU architecture allowed the successful implementation of an otherwise stated as inefficient strategy for the Parallel Simulated Annealing.

References

[1] S. Kirkpatrick , C. D. Gellatt Jr.and M. P. Vecchi Optimization by simulated annealing, 1982 :IBM Thomas J. Watson Research Center
[2] D. B. Kirk and W. W. Hwu Programming Massively Parallel Processors: A Hands-on Approach, 2010 :Morgan Kaufmann
[3] Willemen, R.J.: School Timetable Construction, Algorithms and Complexity, Ph.D. Dissertation. Technische Universiteit Eindhoven (2002)
[4] Graffigne, C. (1992) Parallel annealing by periodically interacting multiple searches: an experimental study. In: R. Azencott (ed.), Simulated Annealing Parallelization Techniques. John Wiley & Sons, New York, NY, pp. 47–79.
[5] J.A. Chandy, S. Kim, B. Ramkumar, S. Parkes, and P. Banerjee. An Evaluation of Parallel Simulated Annealing Strategies with Applications to Standard Cell Placement. IEEE Transactions on Computer Aided Design, 16:398–410, 1997.