10th International Conference of the Practice and Theory of Automated Timetabling
PATAT 2014, 26-29 August 2014, York, United Kingdom

Graphics Processing Unit acceleration of a memetic algorithm for the Examination Timetabling Problem

Vasileios Kolonias · George Goulas · Panayiotis Alefragis · Christos Gogos · Efthymios Housos

Keywords GPU programming · Memetic algorithms · Examination Timetabling

In this paper, we present experimental results from the use of Graphics Processing Units (GPUs) as an accelerator for the solution of the uncapacitated examination timetabling problem (uETP). To test the implementation of the proposed algorithm we use the Toronto datasets as the benchmark set. Details about the uETP problem, the datasets and solution methods can be found in the survey by Qu et al. [1]. GPU implementations of genetic algorithms have performed significant speedups in other application domains [2],[3].

In this work, we use a simple memetic algorithm as the goal of our study was to demonstrate the acceleration possibilities of the GPUs in conjunction with simple algorithmic implementation. We use an array of integers for the direct representation of chromosomes where each cell represents an examination and the corresponding value represents the time slot the examination should be scheduled. For the local search optimization of individual candidate solutions we selected the steepest descend algorithm although it is relative simple to extend the implementation to use more advanced meta-heuristics.

Vasileios Kolonias · George Goulas · Efthymios Housos
University of Patras, Dept. of Electrical and Computer Engineering, Greece
Tel.: +30-2610962436
E-mail: bakoloni@ece.upatras.gr

George Goulas · Panayiotis Alefragis
Technological Educational Institute of Western Greece. Dept. of Computers & Informatics Engineering, Greece

Christos Gogos
Technological Educational Institute of Epirus. Dept. of Accounting and Finance, Greece

As the data transfers to and from the GPU are time consuming, we executed a performance analysis on an initial sequential implementation to determine the most time consuming computational kernels. According to this analysis, the most consuming kernel was the local search improvement of the incumbent solutions. What is interesting is that the kernel execution for each solution is independent and all of them can be performed in parallel by GPU cores. To simplify the implementation, we decided to only delegate the execution of the local search improvement kernel to the GPU and perform the rest of the steps (crossover, mutation and next generation decisions) of the memetic algorithm on the CPU.

The cost function is a weighted sum of the absolute distance in time slots between all examination pairs that have common students. The original weight vector $w = \{-, 16, 8, 4, 2, 1\}$ states that if there is a conflict, i.e. two examinations that have common students are assigned to the same time slot, the solution is illegal and for distances of more than 6 time slots the weight is zero. To achieve an efficient GPU implementation of the cost evaluation we extended the weight vector to $w = \{40000, 16, 8, 4, 2, 1, 0, \ldots, 0\}$, size been the number of the problem's available time slots and dropped conditional statements. This has as a consequence that every absolute distance has a weight so all GPU cores can perform the same operation. This is very important for GPUs as the existence of an "if" statement leads to vast degradation in GPU performance.

Another important pre-processing step was the calculation of the diagonal conflict matrix, where for every examination i, each row R_i contains only conflicts with examinations with indexes $j \geq i+1$. This helped reduce the memory requirements and improved the access complexity during the evaluation phase. The characteristics of the memetic algorithm used are:

- Tournament selection with size 6% of the population size.
- Uniform crossover operator.
- In the mutation operator two random examinations swap their time slots .
- The steepest descent algorithm for local improvement of the incumbent solution. All chromosome genes are traversed in parallel to find the best time slot for every examination. The process of finding the best time slot for each examination is not independent because the cost of each examination should be calculated with the time slots of all the other examinations fixed. Each thread in a block calculates the cost of an examination for each slot and the exam is moved to the time slot with the smallest cost. This process is repeated for all examinations.

Table 1 presents the achieved speedup for different population sizes as well as the quality results of this work and the results of another evolutionary technique used to solve the same problem, the informed genetic algorithm (IGA) [4]. As the average performed speedup is between 17 and 40 times faster compared to the CPU, there is great potential in using GPUs to solve the ETP using memetic algorithms. Our future plans include the selection of more elaborate methods for local improvement of the solutions and the implementation

Dataset	CPU(s)	GPU(s)	S	CPU(s)	GPU(s)	S	This work	IGA
car-f-92	0,497	0,021	23.67	63,14	1,287	49.06	4.74	**4.2**
car-s-91	0,786	0,034	23.12	100,69	2,234	45.07	5.36	**4.9**
ear-f-83	0,090	0,004	22.50	11,504	0,241	47.73	37.04	**35.9**
hec-s-92	0,020	0,001	20.00	2,544	0,059	43.12	**10.83**	11.5
kfu-s-93	0,097	0,006	16.17	12,372	0,323	38.30	**14.09**	14.4
lse-f-91	0,068	0,005	13.60	8,693	0,238	36.53	11.18	**10.9**
rye-s-93	0,162	0,009	18.00	20,791	0,525	39.60	**8.88**	9.3
sta-f-83	0,015	0,001	15.00	1,933	0,064	30.20	**157.06**	157.8
tre-s-92	0,111	0,006	18.50	14,159	0,340	41.64	8.69	**8.4**
uta-s-92	0,644	0,028	23.00	82,283	1,867	44.07	3.93	**3.4**
ute-s-92	0,013	0,001	13.00	1,667	0,064	26.05	**25.05**	27.2
yor-f-83	0,077	0,004	19.25	9,881	0,226	43.72	**38.2**	39.3
Average Speedup			18,82			40.43		

Table 1 Achieved speedup (S) for a single generation of the steepest descend algorithm and quality results

of all the memetic algorithm phases in the GPU. Furthermore, we would like to solve the datasets of the International Timetabling Competition (ITC) with the proposed method and the use of GPUs.

References

1. Qu, R. and Burke, E.K. and McCollum, B. and Merlot, L.T.G. and Lee, S.Y., A survey of search methodologies and automated system development for examination timetabling, Journal of Scheduling, 12, 55-89 (2009)
2. Pospichal, P. and Jaros, J. and Schwarz, J., Parallel genetic algorithm on the CUDA architecture, Proceedings of the 2010 international conference on Applications of Evolutionary Computation - Volume Part I, 442-451. Springer-Verlag, Istanbul, Turkey (2010)
3. Arora, R. and Tulshyan, R. and Deb, K., Parallelization of binary and real-coded genetic algorithms on GPU using CUDA, 2010 IEEE Congress on Evolutionary Computation (CEC), 1-8. Barcelona (2010)
4. Pillay, N.; Banzhaf, W., An informed genetic algorithm for the examination timetabling problem. Appl. Soft Comput. 2010, 10, 457-467