A New Hybrid GPU-PSO Approach for Solving Max-CSPs

Dali Narjess*
COSMOS Laboratory,National School of Computer Science
University of manouba,Tunisia
narjess.dali@ensi-uma.tn

Bouamama Sadok
COSMOS Laboratory,National School of Computer Science
University of manouba,Tunisia
Sadok.Bouamama@ensi.rnu.tn

ABSTRACT

Particle swarm optimization (PSO) has been considered as a very efficient swarm intelligence technique used to solve many problems, such as those related to Constraint reasoning in particular Constraint Satisfaction Problems (CSPs). In this paper, we introduce a new PSO method for solving Maximal Satisfaction Problems Max-CSPs, which belong to CSPs extensions. Our approach is based on a combination between two concepts: double guidance by both template concept and min-conflict heuristic, and the Triggered mutation proposed by Zhou and Tan. This new proposed approach avoids premature stagnation process in order to improve Max-CSPs solution quality. We resort to the high parallel computing insofar as it has shown high performances in several fields, using GPU architecture as a parallel computing framework. The experimental results, presented at the end, show the efficiency of the introduced technique in the resolution of large size Max-CSPs.

1. INTRODUCTION

CSPs [5] have been classified as NP hard problems. They are designed to represent real-world constraint-based problems. Solving a CSP consists in finding an optimal solution which is a complete instantiation of the variables satisfying all the constraints of the problem. Actually, CSPs have multiple extensions defined according to the problem objectives,especially Maximal Constraint Satisfaction Problem (Max-CSP) which is the focus of our work. The resolution of Max-CSPs consists in finding a solution that satisfies the maximum constraints number. Meta-heuristics are among the efficient methods for solving CSPs [1, 2, 4], such as D^3GPSO[1] which is a PSO-based method[3] . The appearance of the high parallel computing, has allowed researchers to exploit the powerful many-cores architectures, such as the Graphical Processing Units (GPU). Many PSO-based approaches using GPU were proposed to significantly accelerate the execution time [7, 2]. In this work, we will present a novel hybrid optimized PSO-based method for solving Max-CSPs using GPU. The proposed technique gives both better solution quality and better execution time results compared to some other PSO based-methods.

*Corresponding author

GECCO '16 July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.
DOI: http://dx.doi.org/10.1145/2908961.2908973

2. CONTRIBUTION

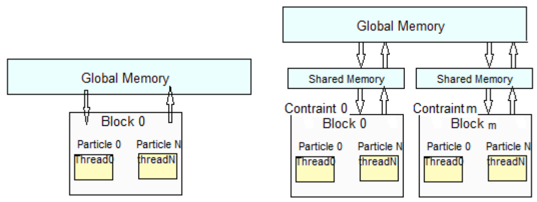
Our proposed approach is a new hybrid PSO-based method using GPU for solving Max-CSPs. It relies on the combination of the double guidance by template concept and min-conflict heuristic used in D^3GPSO [1] as well as on the Triggered mutation and memory mechanism concepts used in PSO-TM [7]. These concepts are explained as follows:

Template concept and min-conflict heuristic: The evaluation of the Fitness values (FVs) is based on template concept [1, 2]. The latter consists in attributing a template to each particle. Each element of the template is called weight. It corresponds to the particle variable. A weight is defined as the sum of the violated constraints in which the related variable is involved. The particle FV is the maximum value of its template. The min-conflict heuristic is applied in order to enhance the particles FVs.

Triggered Mutation concept: You Zhou and Ying Tan developed a new PSO algorithm with Triggered Mutation [7], called PSO-TM : If only less than P % of all the particles update their local best position during S consecutive iteration, then the swarm is in a stagnant status, and a mutation, with a probability R, has to be triggered. Before applying the mutation operator, the velocity and position vectors of the particles have to be memorized, in order to be restored if the FVs of the mutated particles are not better than the previous values before the mutation. We propose two implementations using GPU for our new hybrid PSO-based method:

1- GPU-DGPSO-TM : It is based on one parallelism level, where only PSO is parallelized; the particles are executed in parallel, but in the FV evaluation step, each particle checks sequentially the satisfaction of the constraints. In the CUDA implementation of our first approach, each thread is associated to one particle.See Figure 1a.

2- GPU-DGPSO-CSP: It is based on two parallelism levels. Here, we propose to parallelize not only our proposed double-guided PSO with triggered mutation, but also the Max-CSP; we suggest to parallelize even the previously-mentioned satisfaction with maintaining the particles parallel execution. This approach is designed in a manner that all the constraints are checked at the same time by all the particles that run concurrently. So, both the meta-heuristic and the problem, are parallelized. To ensure these two parallelism levels, each constraint is associated with a CUDA block and each thread of a block is assigned to a particle as presented in Figure 1b.



(a) DGPSO-TM CUDA design (b) DGPSO-TM-CSP CUDA design

Figure 1: Our two proposed approaches CUDA design

3. EXPERIMENTATIONS

Our two proposed approaches are validated on randomly-generated binary CSPs.The generation of a CSP is guided by four parameters which are: variables number (n), domain size (d), density (p) and tightness (q) [1, 2]. The numerical values chosen are the following: n=40, d=40, p and q are taken as 0.1, 0.3, 0.5, 0.7 and 0.9 which gives 25 density-tightness possible combinations. Three main zones are considered from the density values: Easy problems zone for density from 0.1 to 0.3 (zone 1), transition CSPs phase for density values from 0.3 to 0.7 (zone 2), and hard problems zone containing density from 0.7 to 0.9 (zone 3). For the PSO algorithm parameters, the swarm size N=1000 and the number of iterations is equal to 100. For the parameters of the Triggered mutation concept, P = 40, S = 4 and R = 0.2 [7]. We performed three comparisons in terms of execution time and Fitness ratios: **Comparison 1** is made between GPU-DGPSO-TM and GPU-DGPSO. The latter is the counterpart of our GPU-DGPSO-TM but without including the triggered mutation. **Comparison 2** is between GPU-DGPSO-TM and GPU-PSO-TM. It does not include the double guidance by template concept and min-conflict heuristic. This comparison will emphasize the contribution of these two concepts.**Comparison 3** is performed between our GPU-DGPSO-TM and GPU-DGPSO-TM-CSP to show the effect of two parallelism levels.All the approaches are implemented on GPU to solve randomly generated Max-CSPs. The results in terms of Fitness values and Execution time are reported in Table 1. We conclude from comparison 1 and comparison 2 that the hybridization is efficient to improve the Max-CSPs solution quality without troubling the execution time. In fact, the Fitness ratios are superior to 1 in the three zones, and values of more than 2 are obtained in zones 2 and 3. From comparison 3, we notice that our hybrid approach, based on two parallelism levels, is more efficient than the one based on one parallelism level in terms of execution time, while keeping exactly the same solution quality.

Table 1: Fitness(F) and Execution time(T) Comparisons

Zones	**Comparison1**	**Comparison2**	**Comparison3**
zone1	F:1.33 T:1.018	F:1.45 T:1.17	F:1.001 T:1.69
zone2	F:1.82 T:1.009	F:2.08 T:1.15	F:1.03 T:1.707
zone3	F:2.32 T:1.0007	F:2.58 T:1.19	F:1.002 T:1.71

4. CONCLUSION

In our work, we have introduced a new hybrid PSO-based method using GPU for Max-CSPs. This hybridization is a combination between the double guidance by both template concept and min-conflict heuristic and triggered mutation proposed by Zhou and Tan [7]. The double guidance by template concept and min-conflict heuristic has been proven to be efficient for solving CSPs [1]. Now, with the add of triggered mutation concept, we have further improved the CSPs solution quality to attain good results. In fact, we have presented a comparison between our proposed GPU-DGPSO-TM and GPU-PSO-TM, and we have compared it to GPU-DGPSO, inspired from D^3GPSO, which does not use triggered mutation. After that, and for further improvements, we have proposed another method to implement DGPSO-TM on GPU. The new implementation is based on two parallelism levels. In fact, in this approach, GPU-DGPSO-TM-CSP, both the hybrid PSO and the Max-CSP have been parallelized. GPU-DGPSO-TM-CSP has given much better execution time than GPU-DGPSO-TM. So, it would be better to use GPU-DGPSO-TM-CSP when it comes to solving large size Max-CSPs as it maintains the same good solution quality as GPU-DGPSO-TM. No doubt further refinement of our approach would allow improving its performance. We want to thank the reviewers for their wise comments to progress our work.

5. REFERENCES

[1] Bouamama.S: A new distributed particle swarm optimization algorithm for constraint reasoning, Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems: Part II, September 08-10, Cardiff, UK,2010.

[2] Dali.N, Bouammama.S: GPU-PSO : Parallel Particle Swarm Optimization approaches on Graphical Processing Unit for Constraint Reasoning: Case of Max-CSPs. International Conference on Knowledge-Based and Intelligent Information & Engineering Systems,Procedia Computer Science 60, 1070 - 1080,2015.

[3] Kennedy. J, Eberhart.R.C : Particle Swarm Optimization. Proc. of IEEE International Conference on Neural Network, Piscataway, NJ. Pp.. 1942-1948,1995.

[4] Methlouthi.I, Bouamama.S, A family of honey-bee optimization algorithms for Max-CSPs. International Journal of Knowledge-Based and Intelligent Engineering Systems, Vol19(4): 215-224,2016

[5] Tsang.E.P.K: Foundations of Constraints Satisfaction, Academic Press Limited, UK,1993.

[6] Tsang.E.P.K, Wang.C.J, Davenport.A, Voudouris.C,Lau.T.L.: A family of stochastic methods for Constraint Satisfaction and Optimization. Technical report, University of Essex, Colchester, UK,1999.

[7] Zhou.Y, Tan.Y : Particle swarm optimization with triggered mutation and its implementation based on GPU. Proceedings of the GECCO'10, pp. 1007-1014,2010.