SZYMON ŁUKASIK[1,2], KRZYSZTOF BURY[1], KRZYSZTOF JAKUBIK[1]

RÓWNOLEGŁE ALGORYTMY W PEŁNI DOINFORMOWANYCH ROJÓW

PARALLEL FULLY-INFORMED SWARM ALGORITHMS

S t r e s z c z e n i e

Artykuł przedstawia ideę implementacji dwóch popularnych metaheurystyk rojowych: optymalizacji w pełni doinformowanym rojem cząstek i algorytmu świetlika, z użyciem metod przetwarzania równoległego, a w szczególności procesorów graficznych. Poza samym zaprezentowaniem zasad działania obu algorytmów przedmiotem rozważań pracy są procedury wykorzystujące współbieżne przetwarzanie. Praca, poza zilustrowaniem oczywistych zalet takiego podejścia – zwłaszcza w aspekcie wydajnościowym – omawia również w skrócie ograniczenia i słabości algorytmów opartych o obliczenia równoległe.

Słowa kluczowe: optymalizacja w pełni doinformowanym rojem cząstek, algorytm świetlika, przetwarzanie równoległe, metaheurystyki

A b s t r a c t

This article discusses a possibility of implementing two popular swarm-based metaheuristics: Fully Informed Particle Swarm Optimization and Firefly Algorithm, with the use of parallel processing methods – by means of GPUs in particular. Alongside with a brief presentation of the workings of the two algorithms, ideas of employing concurrent processing are under consideration. Besides obvious advantages – in terms of performance – we also shortly discuss limitations and weaknesses of parallel processing approach.

Keywords: fully informed particle swarm optimization, firefly algorithm, parallel processing, metaheuristics

[1] Department of Automatic Control and Information Technology, Cracow University of Technology
[2] Systems Research Institute, Polish Academy of Sciences
Corresponding author e-mail: szymonl@pk.edu.pl

1. Introduction

Particle Swarm Optimization introduced by Kennedy, Eberhart and Shi in 1995 [4] constitute nowadays one of the most important nature-inspired metaheuristics. An idea to represent each solution of the optimization problem at-hand as a member of the swarm – communicating with others and modifying its position under the influence of best individuals – proved to be extremely successful. The degree of this success can be represented by the significant amount of contributions employing PSO in real-world problems e.g. in data analysis [2], resource allocation [8] etc. It can be also quantified through a number of related algorithms, based on the idea of intelligent swarms. One of recent examples of such include: Quantum-behaved Particle Swarm Optimization [3] and multi-swarm PSO [9].

The general goal of unconstrained optimization that is to find x^* which satisfies:

$$x^* = \min_{x \in S} f(x), \tag{1}$$

where $S \subset \mathrm{R}^n$, Initial PSO algorithm's behavior was built on the assumption that each individual member of the swarm, i.e. solution $x_1, x_2, ..., x_m$ of the optimization problem (1), changes its velocity vector in the consecutive algorithm's iteration as a result of the influence of two specific solutions: the best one found so far by the swarm and the top solution identified by this individual.

Fully Informed Particle Swarm Optimization (FIPSO) presented first by Mendes, Kennedy and Neves [7] constitute a modification of this approach. In the most general variant of FIPSO the velocity update is constructed using weighted average position of all swarm members. Creators of the algorithm considered alternative communication topologies e.g. ring or cluster as well as different schemes of assigning weights to prioritize individuals' inputs.

Firefly Algorithm (FA) created by Xin She Yang in 2008 is constructed on similar assumptions [11]. The position of swarm member x_i within feasible solution space S is determined by other individuals' fitness – better solutions will attract those which are worse, in the sense of selected cost function f value [6]. That is why FA – seen as position-based PSO – like FIPSO can be perceived as a member of broader family of techniques named here Fully Informed Particle Swarm algorithms

The goal of this contribution is to discuss a possibility of improving the performance of two aforementioned optimization strategies by employing parallel processing. We will discuss the possible schemes of parallelization and both advantages and weaknesses of proposed approach.

2. Implementing Parallel Fully-Informed Particle Swarms

Swarm optimization techniques belong to population-based metaheuristics. Their parallelization can be achieved in two ways: parallelization of computations, in which the operations commonly applied to each of the individuals are performed in parallel and parallelization of population, in which the population is split into different parts that can be simply exchanged or evolved separately, and then joined later [1].

The approach which is natural for Fully-Informed Particle Swarms is concurrent calculation of cost function values. The amount of processing units P in that case should exceed population size m. It can be achieved by employing modern Graphical Processing Units (GPUs). The illustration of parallelization process used in this study is enclosed on Fig.1.

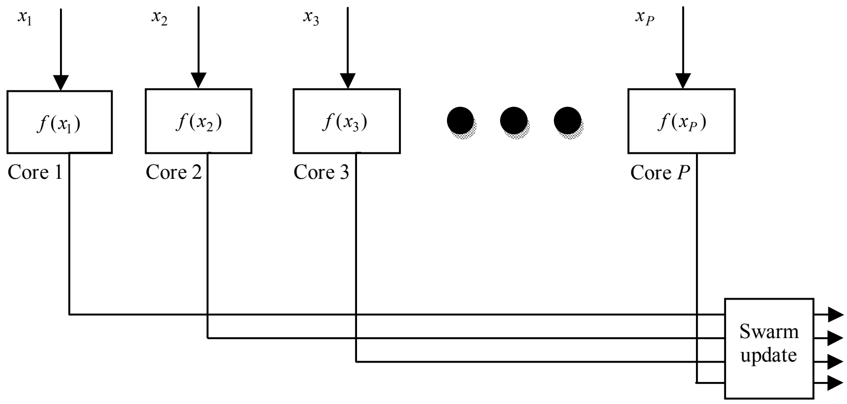

Figure 1. Parallelization of Fully-Informed Particle Swarm algorithms.

Rysunek 1. Zrównoleglenie algorytmów w pełni doinformowanych rojów cząstek.

To evaluate the performance of implementation for both parallel algorithms we have performed extensive experimental studies using five commonly used benchmark functions (namely: *sphere, Rosenbrock, Griewank, Rastrigin* and *Ackley*) using NVIDIA CUDA-enabled graphic card with 336 cores [12]. The actual speed improvement is obviously related to computational effort associated with given cost function. For optimization problems characterized by low dimensionality the benefit of parallelization was found to negligible. It is a direct consequence of additional time delays related to copying memory content, synchronization and data conversion. Nevertheless the improvement for other instances was observed for parallel variants of both – FIPSO and FA.

When discussing parallelization of swarm-based metaheuristics one should also take into account limitations of tools at hand. In case of GPU computing by means of CUDA problems which may arise are relatively slow access to global memory, floating numbers precision and some deviations from programming standards [10]. Therefore special care needs to be taken when using those tools in practical problems of data analysis and optimization.

3. Summary and concluding remarks

The paper presents the possibility of implementing popular swarm-based metaheuristics: Fully Informed Particle Swarm Optimization and Firefly Algorithm in parallel processing environment provided by multiple core GPUs. Although the idea of concurrent of population-based techniques is not new, it is nowadays very hot research area – thanks to massive parallelization possibilities. Further studies in this area will concern supplementary experimental evaluation of created algorithms, as well as the implementation of another fully-informed particle swarm technique – Glowworm Swarm Optimization [5] – in parallel setting.

Acknowledgements

The study is co-funded by the European Union from resources of the European Social Fund. Project PO KL “Information technologies: Research and their interdisciplinary applications”, Agreement UDA-POKL.04.01.01-00-051/10-00.

References

[1] Alba, E., Luque, G., Nesmachnow, S., *Parallel metaheuristics: recent advances and new trends*, International Transactions in Operational Research, vol. 20, 2013, 1–48.

[2] Cura, T., *A particle swarm optimization approach to clustering*, Expert Systems with Applications, vol. 39/1, 2012, 1582–1588.

[3] Fang, W., Sun, J., Ding, Y., Wu, X., Xu, W., *A Review of Quantum-behaved Particle Swarm Optimization*, IETE Technical Review, vol. 27, 2010, 336–348.

[4] Kennedy, J.; Eberhart, R., *Particle Swarm Optimization*, Proceedings of IEEE International Conference on Neural Networks, 1995, vol. IV, 1942–1948.

[5] Krishnanand, K.N., Ghose, D., *Glowworm Swarm Optimization for Simultaneous Capture of Multiple Local Optima of Multimodal Functions*, Swarm Intelligence, vol.3/2, 2008, 87–124.

[6] Łukasik, S., Żak, S., *Firefly Algorithm for Continuous Constrained Optimization*, Lecture Notes in Artificial Intelligence, vol. 5796, 2009, 97–106.

[7] Mendes, R., Kennedy, J., Neves, J., *The Fully Informed Particle Swarm: Simpler, Maybe Better*, IEEE Transactions on Evolutionary Computation, vol.8/3, 2004, 204–210.

[8] Morsly, Y., Aouf, N., Djouadi, M.S., Richardson, M., *Particle Swarm Optimization Inspired Probability Algorithm for Optimal Camera Network Placement*, IEEE Sensors Journal, vol.12/5, 2012, 1402-1412.

[9] Röhler, A.B, Chen, S., *Multi-swarm hybrid for multi-modal optimization*, Proceedings of the IEEE Congress on Evolutionary Computation, 2012, 1759-1766.

[10] Shams, R., Barnes, N., *Speeding up Mutual Information Computation Using NVIDIA CUDA Hardware*, Proceedings 9th Biennial Conference of the

Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, 2007, 555–560.
[11] Yang, X. S., *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Frome, 2008.
[12] Nvidia™, *Nvidia CUDA C Programming Guide, ver. 4.0*, 2011.