International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Survey of Genetic Algorithm Approach for Nurse Scheduling Problem

Swapnaja S. Balekar[1], N. A. Mhetre[2]

[1]ME Student, Department of CE, Smt. Kashibai Navale College of Engineering, Vadgaon, MS, India.

[2]Assistant Professor, Department of CE, Smt. Kashibai Navale College of Engineering, Vadgaon, MS, India

Abstract: ***Nurse Scheduling is a complex task that arises in everyday activities at hospitals system. Most of the scheduling problems are NP-hard. The Nurse Rostering Problem [NRP] is a subclass of the personnel scheduling problems. As nurse scheduling done manually and requires much time, there is need to provide solution to automate the process of scheduling. This paper provides the information about various methodologies for solving NRP and tells NRP can be solved by GA and PGA with the help of review.***

Keywords: GPGPU (General Purpose Graphics Processing Units), Heuristics, Nurse Rostering Problem(NRP), Genetic Algorithm, Parallel Genetic Algorithm

1. Introduction

Nurse roster is nothing but a weekly or monthly plan for all nurses in hospital, and is obtained by assigning shift categories to the nurses. Nurse Scheduling represents a task which consists of creating a schedule for the nurses in a hospital. The Nurse Rostering Problem (NRP) is a common problem every hospital faces every day. NRP belongs to personal scheduling problems and requires to provide an optimal schedule based on the working hours of the nurses, their personal choices to shift types, hospital rules and government laws. In other words, this problem is stated as follows: assign shifts to nurses for a certain time period which must subject to satisfy a set of constraints. Generally, constraints are put forward by regulations, working practices and the preferences of the nurses. NRP represents an interesting field of research and development. Osogami and Imai[70] (2000) proved that the nurse rostering problem is NP-hard. In fact, they proved that the timetabling problem, which is NP-complete, can be transformed into a decision version of the nurse rostering problem with only a subset of the real world constraints that applied to it.

Genetic Algorithms have proved to be an efficient algorithm for finding near optimal solution to a scheduling problem. Nowadays, this process is done manually by highly qualified medical administrator and thus requiring time and involvement of the personnel. The primary reason for this is that hospitals are operational, 24 hours a day, 7 day a week. Getting a solution to automate the schedule, represents a challenge to both Operations Research and Artificial Intelligence communities and the Personal Scheduling communities. Constraint Programming is an approach which can be used to model the constraints given by the hospital labor policy as well as the government legal regulations.

There is need to develop an efficient algorithm for generating for the nurse timetable. Obtaining an optimal schedule in shorter time is another goal. Nowadays, schedule is done by the computer, and the person gets involved in creating roster, not available for providing health care demands. If nurse roster is static, it cannot handle the dynamic environment of the hospital, where employees might take days off on a short notice, or there can be a sudden and unexpected requirement for a higher number of nurses available [2].

This paper presents overview of many methodologies available to solve the NRP and gives the idea of parallelism for solving NRP with the help of GPGPU. In this paper, section 2 gives information about constraints and gives problem definition for the NRP and in Section 3, we describe the solution approaches that are used and available for this problem. In Section 4, we discuss the idea of parallelism with heuristics for solving this problem and in Section 5 we provide some conclusions over this review.

2. Problem Formulation

NRP is formulated with the help of types of constraints and the types of a problem listed below.

1) Constraints

- Hard constraints: Feasible roster is obtained when hard constraints are met. These constraints describe a combination of law and hospital requirements that must be enforced upon the roster [6].
- Soft Constraints: Roster quality depends on the soft constraints satisfaction. These are designed to improve the actual roster. Because not only usable roster, but also a satisfied workforce is needed to meet the high quality care demands. These constraints may vary. These constraints include requests for free days, shift type preferences or requests for longer free time blocks between worked shifts [6].

The goal is always to schedule resources to meet the hard constraints while aiming at a high quality result with respect to soft constraints [2]. Some of the Hard and Soft constraints are listed in table I.

Constraints Types

HC1: A nurse may not start more than one shift each day

HC2: The number of assigned nurses has to match the exact demand
HC3: A nurse must match the skills required for the shifts they work
SC1: Complete weekends
SC2: Minimum consecutive Free days
SC3: Maximum consecutive Free days
SC4: Maximum number of shifts in planning period
SC5: No night shift before free weekends
SC6: Maximum shift type in week
SC7: Minimum time between two shifts
SC8: Avoiding certain shift successions
SC9: Maximum consecutive working days
SC10: Maximum working weekends
SC11: Maximum hours worked per nurse
SC12: Number consecutive shifts in planning
SC13: Two free days after series of night shift
SC14: Maximum shift type in planning period
SC15: Min Working days
SC16: Same shift type for weekend
SC17: Requested day-off
SC18: Bank holidays
SC19: Alternative skill
SC20: Max Shift Day of Week

2) Problem type
Depending on constraints, a NRP falls into Optimization Problem or a Decision problem [1].

Optimization problem: In this approach, the problem was formulated to minimize or maximize an objective function. Mathematical Programming is proved as an exact approach to combinatorial Optimization Programming. Traditional methods from linear programming, integer programming, GP networks have also been employed to solve the NRP.

Decision Problem: In NRP, large number of constraints has to be checked, so it can be more appropriate to model the NRP as a constraint satisfaction Problem (CSP). Feasible solutions to the CSP are nothing but the assignments of values to variables satisfying all constraints. Decision problems are solved by Heuristics or AI.

3. Solution Approaches

Cyclic and non-cyclic scheduling are the two types of scheduling that are used for NRP. Cyclic scheduling has repeated pattern in consecutive scheduling periods, whereas in non-cyclic scheduling, a new schedule is generated for each scheduling period. Cyclic scheduling was first used in the early 1970s which has low computational requirements and the greater possibility for manual solution. The algorithms for the NRP generally use cyclic scheduling.

Solution approaches for the NRP can be classified into two main categories: The optimization approach and the decision approach. The optimization approach is usually based on MP techniques, while the decision approach usually based on heuristics and other AI tools [1].

A. Mathematical Model programming: Earlier methods were based on mathematical programming, started early in 70's. These methods often provided a guarantee for reaching the absolute optimum, but they do not perform well in real situations. Search spaces for the real NRPs are very big [1]. In [3]-[5] authors have given detailed description of Mathematical Programming techniques.

B. Goal Programming: It is used as an improvement in the mathematical approaches, since they can often only optimize one single goal [2]. Goal programming is used to solve scheduling problems and also worked well optimizing the solutions. [6]-[10].

C. Constraints Programming: CP provides a powerful tool for finding feasible solutions to rostering problems. It is useful if the problem is highly constrained and/or when any feasible solution will suffice even if it is not optimal. This technique doesn't produce good solutions for problems where the main challenge is to find an optimal or near optimal solution out of a vast number of feasible solutions [1]. Constraint logic programming languages described constraint logic easily. Constraint programming is applied for scheduling problem [11]-[16].

Meta-heuristics GAs has been used for solving the NRP, (for example [20]-[25], [27], [29], [32], [33], [35]). Sequential GAs have also proved very successful in many applications and in very different domains. Genetic Algorithms (GAs) are efficient search methods based on principles of natural selection and genetics. They are being applied successfully to find acceptable solutions to problems in business, engineering, and science [33]. GAs are generally able to find good solutions in reasonable amounts of time, but as they are applied to harder and bigger problems there is an increase in the time required to find adequate solutions[35]. As a consequence, there have been multiple efforts to make GAs faster, and one of the most promising choices is to use parallel implementations over GPGPU. GA works in context of NRP for crossover and mutation, the best personal schedule from each of the parents schedule can be selected, a random selection from the personal schedule of parents can be selected, or we can select the best events in a schedule. Best solutions in each generation are kept and others are replaced by newly formed solutions [1]. Kawanaka et. al. in [20] used GA to obtain optimal nurse schedules satisfying absolute and desirable constraints. Aickelin et. al. in [21] proposed an indirect method of GA for solving NRP. In [report 4] Author proposed an effective mutation operator for the cooperative GA, which does not affect validity of the schedule. The cooperative GA with the crossover and new mutation operators can give a better schedule than cooperative GA when used with the crossover operator. Author included new constraints like affinity between nurses, Prohibition of assignment of two or more new faces to night duty.

4. Proposed Genetic Algorithm Approach

Genetic Algorithm have shown excellent search abilities, but often lose their efficacy when applied to large and complex problems because a lot of candidate solutions must be evaluated, Many optimization methods suffer from the curse of dimensionality, which shown that their performance decreases quickly when the dimensionality of the search space increases. So, there is a need to provide parallelism in traditional approach. Nowadays, GPGPUs are able to provide the computational resources to handle these high-dimensional problems while maintaining a limited execution time and a high portability [63]. Fortunately, the most time-consuming fitness evaluations can be performed independently for each individual in the population. Genetic algorithm can be parallelized and fitness can be calculated on GPU by using various types of parallelization models like master slave model, fine grained model, island model etc.

A. Genetic Algorithm

A. J. Umbarkar et, al. in [41] provides the review about how various authors, researchers, scientists those have applied GA/PGA on GPGPU with parallelism. Pablo Vidal, Enrique Alba in [42] implemented cellular Genetic algorithm on Multi GPU, and have obtained good result after comparing with CPU and one GPU. Petr Pospichal, Jiri Jaros, and Josef Schwarz in [43] have mapped the parallel island based genetic algorithm with unidirectional ring migrations to nVidia CUDA software model which clearly showed that GPUs have a potential for acceleration of GAs and allow solving the much complex tasks. The results also showed that the proposed GPU implementation of GA can provide better results in the shorter time or can produce better results in equal time. Mihai Calin et. al. in [44] Proposed Genetic algorithm on CUDA for solving NP complete problem. Mohamed Wahib and Asim Munawar in [45] provides a study on adapting legacy parallel GAs on GPGPU systems, reviewed design issues in GPU relevant to parallel GAs. Petr Pospichal, Jiri Jaros in [46] showed that GPU's have proven their abilities for acceleration of genetic algorithms. Impressive speedups were achieved, and also high quality solutions were met. They used nVidia GPU supporting ShaderModel 4.0 and Linux/Windows platform for analysis. In [57] parallel results are compared with the sequential algorithm on accuracy and clock time for varying problems by studying the effect of a number of parameters, namely population sizes, number of threads, problem sizes, and problems of differing complexities. Researchers of this paper have gained better results in every parameter criteria. In [58] author presents parallelization of the OX (order crossover) operator and experimentally showed that parallelized OX crossover operator is effective on a GPU based on the CUDA architecture. Author practiced with an NVIDIA GeForce GTX580 GPU show that GPU program for the traveling salesman problem (TSP) is about 101.3 times faster than the corresponding CPU program on a single core of 2.67 GHz Intel Xeon X5550. An agent-based scheduling approach [59] is extended with parallel genetic algorithms (PGA) to provide the required optimization support. Test results for PGA have shown better remarks for generating schedules in short time with respect to the predefined set of manufacturing objectives. The extended approach fulfils both flexibility and efficiency requirements on manufacturing scheduling. The analysis of experiment results of the parallel genetic algorithms for Optimization of Modular Neural Networks for Pattern Recognition Using a Cluster of Computers With a Master-Slave Topology, lead us, to see clearly the importance of using several processors to solve this type of problems to achieve fast results [60]. In [61] proposed implementation executes all genetic operations in a generation of the MGG(Minimal Generation Gap) model in a single kernel function. First, by a kernel function call from the host, an SM receives two individuals (parents) from the population in the global memory. Then, all processes such as random number generator, crossover, mutation, sorting, and selection are executed in the SM. Finally, the two selected individuals are sent back to the global memory, and the routine is immediately repeated until the termination criterion is satisfied. Author used Random Number Generator (RNG) because CUDA libraries do not include random number generator functions. Bitonic sort is used for sorting population because other sort can't be easily parallelized in CUDA. For evaluation purpose author checked GPU and CPU computation with the four optimization function.

5. Conclusion and Future Work

After reviewing various papers in area of Nurse Rostering and Genetic Algorithm on GPGPU following possibilities can be considered-

- To achieve good result for the penalty value in literature by applying sequential GA.
- GPGPU is good option for speedup to solve combinatorial problem.
- To compare results of sequential and parallel GA with different performance parameters.
- Based on complexity of problem, search space, it is possible to provide diversity in search space using Genetic Algorithm on GPGPU.
- It is possible to solve Nurse Rostering Problem effectively using Parallel Genetic Algorithm on GPGPU.

References

[1] Cheang, B., Li, H., Limand, A., & Rodrigues, "Nurse Rostering Problem – a bibliographic survey", European Journal of Operational Research, 151(3), 447–460.
[2] Brno, "Master Thesis-Nurse rostering", Masaryk University Faculty of Informatics- 2009.
[3] Warner, M. (1976), "Scheduling nursing personnel according to nurse preference. A mathematical programming approach", Operations Research, 24(5), 842–856.
[4] E. Yilmaz, "A mathematical programming model for scheduling of nurses labor shifts", Journal of Medical Systems 36 (2012) 491–496.
[5] P.R. Harper, N.H. Powell, J.E. Williams, "Modeling the size and skill-mix of hospital nursing teams", Journal of the Operational Research Society 61 (2010) 768–779.
[6] S. Zolfaghari, Q. Vinh, A. El-Bouri, M. Khashayardoust, "Application of a genetic algorithm to staff scheduling in retail sector", International Journal of Industrial and Systems Engineering 5 (2010) 20–47.
[7] S. Topaloglu, I. Ozkarahan, "An implicit goal programming model for the tour scheduling problem considering the employee work preferences", Annals of Operations Research 128 (2004) 135–158.
[8] L. Hung-Tso, C. Yen-Ting, C. Tsung-Yu, L. Yi-Chun, "Crew rostering with multiple goals: an empirical study", Computers & Industrial Engineering 63 (2012) 483–493.

[9] M.N. Azaiez, S.S. Al Sharif, "A 0-1 goal programming model for nurse scheduling", Computers & Operations Research 32 (2005) 491–507.
[10] J. Clerk Maxwell, "A Treatise on Electricity and Magnetism", 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp. 68-73.
[11] E.K. Burke, J.P. Li, R. Qu, "A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems", European Journal of Operational Research 203 (2010) 484–493.
[12] R. Cipriano, L. Di Gaspero, A. Dovier, "Hybrid approaches for rostering: a case study in the integration of constraint programming and local search", Hybrid Meta heuristics, Lecture Notes in Computer Science, vol. 4030, 2006, pp. 110–123.
[13] F. He, R. Qu, "A constraint programming based column generation approach to nurse rostering problems", Computers & Operations Research 39 (2012) 3331–3343.
[14] J.P. Metivier, P. Boizumault, S. Loudni, "Solving nurse rostering problems using soft global constraints", in: 15th International Conference on Principles and Practice of Constraint Programming (CP 2009), Lisbon, Portugal, Lecture Notes in Computer Science, vol. 5732, 2009, pp. 73–87.
[15] R. Qu, F. He, "A hybrid constraint programming approach for nurse rostering problems", in: SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, England, 2009, pp. 211–224.
[16] L. Trilling, A. Guinet, D. Le Magny, "Nurse scheduling using integer linear programming and constraint programming", 12th IFAC International Symposium, vol. 3, Elsevier, 2006, pp. 651–656.
[17] A.H.W. Chun, S.H.C. Chan, G.P.S. Lam, F.M.F. Tsang J. Wong, "Nurse rostering at the hospital authority of Hong Kong", AAAI/IAAI 2000, pp. 951–956.
[18] S. Abdennadher, H. Schlenker, "Nurse scheduling using constraint logic programming", AAAI/IAAI, 1999, pp. 838– 843.
[19] R.J. Wallace, "Constraint Programming and Large Scale Discrete Optimization", DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 57, DIMACS, 2001, pp. 67–99.
[20] Kawanaka, H., Yamamoto, K., Yoshikawa, T., Shinogi, T., and Tsuruoka, S. "Genetic Algorithms with the Constraints for Nurse Scheduling Problem", Proc. of IEEE Congress on Evolutionary Computation (CEC), Seoul (2001) 1123-1130
[21] Aickelin, U., and Dowsland, K, "An Indirect Genetic Algorithm for a Nurse Scheduling Problem". Computers & Operations Research, 31(5) (2003) 761-778
[22] U. Aickelin, K. Dowsland, "Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem", Journal of Scheduling 3 (3) (2001) 139–153.
[23] U. Aickelin, P. White, "Building better nurse scheduling algorithms", Annals of OR, submitted for publication.
[24] Master thesis, "Solving the Nurse Rostering Problem" Ilina Stoilkovska European Master in Computational Logic Technische, University at Dresden. January 6, 2013
[25] Burke, E.K., Cowling, P.I., De Causmaecker, P. and Vanden Berghe, G., "A Memetic Approach to the Nurse Rostering Problem", 2001, Applied Intelligence, vol 15, pp. 199-214.
[26] Cowling, P.I., Kendall, G., Han, L., "An Investigation of a Hyperheuristic Genetic Algorithm Applied to a Trainer Scheduling Problem". Accepted for 2002 Congress on Evolutionary Computation (CEC2002), Hilton Hawaiian Village Hotel, Honolulu, Hawaii, May 12-17, 2002"
[27] R. Bai, E.K. Burke, G. Kendall, J. Li, B. McCollum, "A hybrid evolutionary approach to the nurse rostering problem", Transactions on Evolutionary Computation 14 (2010) 580–590.
[28] G.R. Beddoe, S. Petrovic, "Selecting and weighting features using a genetic algorithm in a case-based reasoning approach to personnel rostering", European Journal of Operational Research 175 (2006) 649–671.
[29] E.K. Burke, T. Curtois, G. Post, R. Qu, B. Veltman, "A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem", European Journal of Operational Research 188 (2008) 330–341.
[30] L. Frey, T. Hanne, R. Dornberger, "Optimizing staff rosters for emergency shifts for doctors", in: 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, vols. 1–5, 2009, pp. 2540–2546.
[31] J. P. Li, U. Aickelin, E.K. Burke, "A component-based heuristic search method with evolutionary eliminations for hospital personnel scheduling", Informs Journal on Computing 21 (2009) 468–479.
[32] B. Maenhout, M. Vanhoucke, "An evolutionary approach for the nurse rerostering problem", Computers & Operations Research 38 (2011) 1400–1411.
[33] M. Moz, M.V. Pato, A genetic algorithm approach to a nurse rerostering problem, Computers & Operations Research 34 (2007) 667–691.
[34] J. Puente, A. Gomez, I. Fernandez, P. Priore, "Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms", Computers & Industrial Engineering 56 (2009) 1232–1242.
[35] C. C. Tsai, S. H. A. Li, "A two-stage modeling with genetic algorithms for the nurse scheduling problem", Expert Systems with Applications 36 (2009) 9506–9512.
[36] S. Zolfaghari, Q. Vinh, A. El-Bouri, M. Khashayardoust, "Application of a genetic algorithm to staff scheduling in retail sector", International Journal of Industrial and Systems Engineering 5 (2010) 20–47.
[37] Walter J. Gutjahra, Marion S. Raunerb, "An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria", Computers & Operations Research 34, ELSEVIER 2007.
[38] Jau-Ming Su, Jen-Yu Huang, "Using Ant Colony Optimization to Solve Train Timetabling Problem of Mass Rapid Transit"
[39] Leopoldo Altamirano, Maria-Cristina Riff, "A PSO algorithm to solve a Real Anesthesiology Nurse Scheduling Problem", IEEE 2011.
[40] Frederic Pinel, Bernabé Dorronsoro, Pascal Bouvry, "Solving very large instances of the scheduling of independent tasks problem on the GPU" J. Parallel Distrib. Comput. 2012.

[41] A. J. Umbarkar, “GAs on GPGPU: Parallelism Review”, First National Conference on Algorithms and Intelligent Systems, 03-04 February, 2012
[42] P. Vidal and E. Alba, “A multi-GPU implementation of a cellular genetic algorithm”, IEEE explorer, 2010.
[43] Petr Pospichal, Jiri Jaros, and Josef Schwarz, “Parallel Genetic Algorithm on the CUDA Architecture”, Verlag Berlin Heidelberg Springer-2010.
[44] Mihai Calin et. al., “Solving NP complete Problem on CUDA architecture using Genetic Algorithm”, IEEE-2011.
[45] Mohamed Wahib and Asim Munawar, “Optimization of Parallel Genetic Algorithms for nVidia GPUs”, IEEE-2011.
[46] Petr Pospichal, Jiri Jaros, “GPU-based Acceleration of the Genetic Algorithm”, IEEE-2010.
[47] D. Patterson, “The top 10 innovations in the new nvidia fermi architecture, and the top 3 next challenges,” Parallel Computing Research Laboratory (Par Lab), U.C. Berkeley, Tech. Rep., September 2009
[48] M. L.Wong, “Parallel multi-objective evolutionary algorithms on graphics processing units,” in GECCO ’09: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference. ACM, 2009, pp.2515–2522.
[49] A. Munawar, M. Wahib, M. Munetomo and K. Akama, “Advanced genetic algorithm to solve MINLP problems over GPU”, IEEE explorer, pp. 318-325, 2011.
[50] Yongzhen Ke, Yuhao Li, Dandan Li, “Image Matching using Genetic Algorithm on GPU ”, IEEE explorer, 2011.
[51] M. Oiso and Y. Matumura, “Accelerating Steady-state genetic algorithms based on CUDA architecture”, IEEE explorer, pp. 687-692 2011.
[52] M. Wong, T. Wong, and K. Fok, “Parallel evolutionary algorithms on graphics processing unit”, in IEEE Congress on Evolutionary Computation, 2005., Sept. 2005, vol. 3, pp. 2286–2293
[53] J. Li, X. Wang, R. He, Z. Chi, “An Efficient Fine-grained Parallel Genetic Algorithm Based on GPU-Accelerated”, in 2007 IFIP International Conference on Network and Parallel Computing - Workshops, IEEE explorer, pp. 855-862, 2007.
[54] S. Debattisti, N. Marlat, L. Mussi, S. Cagnoni, “Implementation of a simple genetic algorithm within the CUDA architecture”, GECCO 2009.
[55] R. Arora, R. Tulshyan, K. Deb, “Parallelization of binary and real coded genetic algorithm on GPU using CUDA”, IEEE explorer, 2010.
[56] S. Tsutsui and N. Fujimoto, “Solving quadratic assignment problems by genetic algorithms with GPU computation: a case study”, in GECCO ’09: Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference, New York, USA, 2009, pp. 2523– 0.
[57] Ramnik Arora, Rupesh Tulshyan, Kalyanmoy Deb, “Parallelization of Binary and Real-Coded Genetic Algorithms on GPU using CUDA”, IEEE 2010,pp 978-1-4244.
[58] Noriyuki Fujimoto, Shigeyoshi Tsutsui , “Parallelizing a Genetic Operator for GPUs”, IEEE Congress on Evolutionary Computation, June 20-23, 2013.
[59] Ghada Abaza , Iman Badr , Peter Goehner and Sabina Jeschke, “Extending an Agent-Based FMS Scheduling Approach with Parallel Genetic Algorithms ”, IEEE 2010.
[60] F. Valdez, P. Melin and H. Parra, “Parallel Genetic Algorithms for Optimization of Modular Neural Networks in Pattern Recognition”, IEEE 2011.
[61] Masashi Oiso et. al., “Accelerating Steady-State Genetic Algorithms based on CUDA Architecture”, IEEE-2011.
[62] Makoto Ohki, Hideaki Kinjo, “Nurse Scheduling by Using Cooperative GA with Efficient Mutation and Mountain-Climbing Operators”, Springer -2010.
[63] H. Bai, D. Ouyang, X. Li, L. He, and H. Yu, “MAX-MIN ant system on GPU with CUDA”, Int. Conf. on Innovative Computing, Information and Control (ICICIC), Dec. 2009, pp. 801–804.
[64] Nicholas A. Sinnott-Armstrong, Casey S. Greene, and Jason H. Moore, “Fast genome-wide epistasis analysis using ant colony optimization for multifact or dimensionality reduction analysis on graphics processing units”, Proceedings of the 12th annual conference on Genetic and evolutionary computation, GECCO 2010, pages 215–216, New York, NY, USA, 2010. ACM.
[65] Fu, J., Lei, L., Zhou, G., “A parallel ant colony optimization algorithm with GPU-acceleration based on all-in roulette selection”, In Proceedings of the 3rd International Workshop on Advanced Computational Intelligence, IWACI 2010, pp. 260–264.
[66] A. Delevacq, P. Delisle, M. Gravel, and M. Krajecki, “Parallel ant colony optimization on graphics processing units”, Journal of Parallel and Distributed Computing, page doi:10.1016/j.jpdc. 2012.01.003, 2012.
[67] M. Randall and A. Lewis, “A parallel implementation of ant colony optimization”, Journal of Parallel and Distributed Computing, 62(9):1421–1432, 2002.
[68] José M. Cecili a, Andy Nisbet b, Martyn Amosb, Manuel Ujaldón, “Enhancing data parallelism for Ant Colony Optimization on GPUs”, Journal of Parallel and Distributed Computing, Elsevier-2013.
[69] Frederic Pinel, Bernabé Dorronsoro, Pascal Bouvry, “Solving very large instances of the scheduling of independent tasks problem on the GPU”, J. Parallel Distrib. Comput. 2013.
[70] T. Osogami and H. Imai, “Classification of various neighborhood opera-tions for the nurse scheduling problem,” in Proc. Algo. Comput., 2001, vol. 1969, pp. 72–83.