ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (78-83), Month: July - September 2015, Available at: www.researchpublish.com

Parallelizing Genetic Operators for Solving Nurse Scheduling Problem on GPGPU

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

[1]ME student, Dept. of CE, Smt. Kashibai Navale College of Engg, Vadgaon, MS, India
[2]Assistant Professor, Dept. of CE, Smt. Kashibai Navale College of Engg, Vadgaon, MS, India

Abstract: **Nurse scheduling is a difficult problem that every hospitals faces every month or a week. It impacts the health care system in the hospitals. The Nurse Rostering Problem [NRP] said to be as a subclass of the personnel scheduling problems, and most of its instances are not solvable. Solving the Nurse Rostering Problem has been research field from many years, though, Nurse Rostering is done manually. 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), Constraint Programming, Soft Constraints, Hard Constraints, Genetic Algorithm, Heuristics, Nurse Rostering Problem, Parallel Genetic Algorithm.**

I. INTRODUCTION

The NRP involves creating a periodic roster for duties of the nurses. These rosters may be planned as per requirement (weekly or monthly). The NRP is commonly contains a nurse's day timetable, a nurse-task plan or nurse-time slot and a shift patterns for nurses. Constraints that are associated with NRPs are decided by hospitals and nurses choices. These constraints can be divided into two categories i.e. hard constraints and soft constraints. Hard constraints commonly include coverage constraints (for example, staff demand matching to skills category per day per shift type) while soft constraints commonly covers constraints like nurses preferences to particular shift type, also cyclic or no-cyclic shifts in a week or a month. NRP always aims to schedule person resources to meet the hospital requirements [1].

Approaches to nurse rostering can be classified in different ways. Burke et al. (2004), offered a classification based upon the model that was developed. Common models are simple, that are nothing but assignment of morning, late or night shifts to a group of equally skilled full time nurses over a limited period (Bellanti et al., 2004), and some were much complex and has addressed many needs considering nurse's shifts, regulations at work,(Meyer auf'm Hofe,2001). Nurse rostering problems are often described as optimization problems. Many different objective functions have been studied that depended on the healthcare and government regulations and laws of labor and even on the ward. Objectives for nurse rostering include one single objective or multiple objective or combination of the following objectives: minimizing the number of constraint violations, minimizing the number of nurses, minimizing overtime, maximizing the coverage, maximizing satisfaction of personal preferences, etc [2]. However, we may find that the developed rosters have not completed or covered complex real-world situations. Each new paper introduces a particular set of constraints and objective function different than the literature. So, it is difficult to compare results with previous research work. Currently, there is Web page which provides a collection of 13 employee timetabling problems, mostly these include nurse rostering ones, this was generated by (Burke et al, 2007, http://www.cs.nott.ac.uk/~tec/NRP/index.html).

These scheduling instances have been derived from real world problems and are available in XML format. There are two approaches for solving nurse rostering problem and that are 1) ANROM (Advanced Nurse Rostering Model) 2) The Cabarost (Case-based Reasoning Rostering).

An efficient algorithm must be developed for the nurse roster and it must obtain an optimal schedule in shorter time. Nowadays, schedule is done by the computer, and the person gets involved in creating roster, which affects the health care quality. However, a disadvantage of such a nurse roster is that it is static, and 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 Genetic Algorithm to solve the NRP and gives the idea of Genetic Algorithm parallelism with heuristics in GPGPU for solving NRP. In Section 2 we have given information about constraints and gives problem definition for the NRP and in Section 3, we describe the solution approaches that are 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 to this review.

II. PROBLEM FORMULATION

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

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 below.

$$\min f(x) = \sum_{i=1}^{i=k} \sum_{s=1}^{s=n} c_s . g_s(x)$$

s = index of soft constraint

n = number of soft constraints

i = index of nurse

k = number of nurses

c_s= *Penalty weight for violation of soft constraints*

$g_z(x)$ = *total number of violations for the soft constraints in solution roster* x.

TYPE OF CONSTRAINTS [9]

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

1) Problem type:

Depending on constraints, a NRP generally can be classified as Optimization Problem or a Constraint Optimization Problems [1].

***Optimization problem*:** The problem was formulated to minimize or maximize an objective function. Mathematical Programming is an exact approach to combinatorial Optimization Programming.

Constraint optimization problems: Real-life applications, if we never use any solution, but consider a good solution. The quality of solutions is determined by single or multiple criteria which are usually incorporated into an objective function. The aim is to get a solution which is able to maximize or minimize the value of the objective function. This is referred to as a constraint satisfaction optimization problem (CSOP), which is a problem that consists of a standard CSP together with an objective function [1].

III. SOLUTION APPROACHES

A 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 have the advantage of describing constraint logic easily. Constraint programming is applied for scheduling problem [3]-[8].

I however, decided to focus mainly upon a class of algorithms known a genetic algorithm. Their use is popular in the field of timetabling and rostering and they have been used successfully in the past. I also found their flexibility and operation intriguing, and wanted to explore it further in this project.

Genetic Algorithm:

Step 1: Encoding of solutions and initialization of population.

Step 2: Fitness evaluation of each individuals.

Step 3: While Stop Criteria not met

i) Reproduction

ii) Crossover

iii) Mutation.

iv) Evaluate Fitness in the newly created population.

End while.

Unlike other heuristic optimization methods which in most cases build upon the ideas of local search, genetic algorithms maintain a population of solutions.

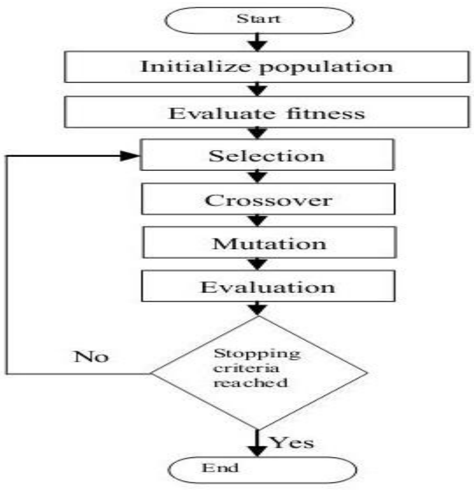

Fig. Flowchart of Genetic Algorithm [12]

Using mutation and crossover operators new offsprings (solutions) are created from the current population. Populations are interchanged following the selected population model until the specified ending criterion is met.

The population of solutions forces the genetic algorithm to be more computationally complex, when compared to other heuristic methods. On the other hand a population of solutions also gives the genetic algorithm better robustness, which turns out to be a advantage especially when dealing with problems where the fitness function contains many local optima[13]-[15].

IV. PROPOSED PARALLEL APPROACH

In PGA, all the steps of Genetic Algorithm are performed but the fitness function evaluation is performed on GPU and results are sent back to GPGPU. Figure below shows the flowchart of parallel genetic algorithm.

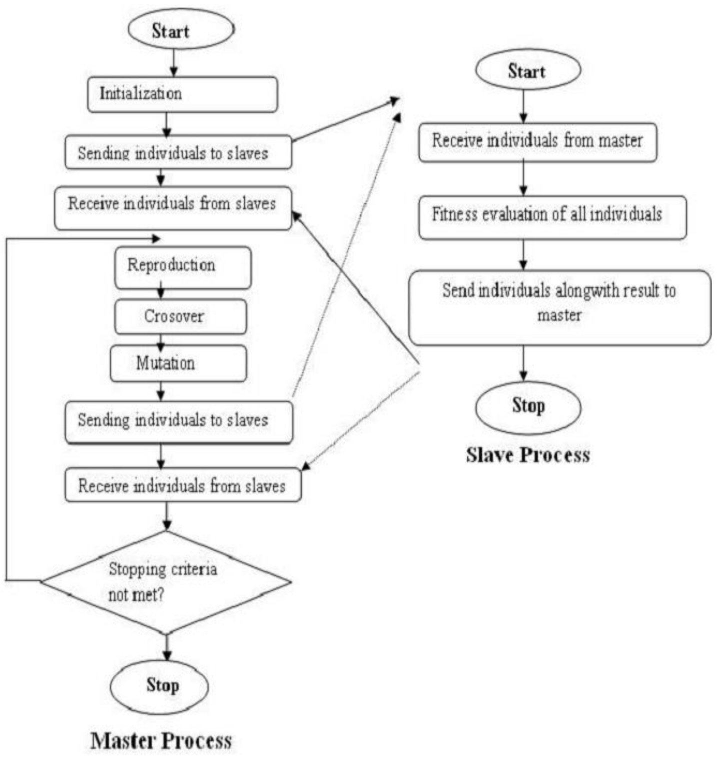
Start
Initialization
Sending individuals to slaves
Receive individuals from slaves
Reproduction
Crosover
Mutation
Sending individuals to slaves
Receive individuals from slaves
Stopping criteria not met?
Stop
Master Process
Start
Receive individuals from master
Fitness evaluation of all individuals
Send individuals alongwith result to master
Stop
Slave Process

Fig. Flowchart Of Parallel Genetic Algorithm

In [16] paper, author has proposed an implementation of steady-state GA on a GPU using CUDA, and also adopted the data parallelization on GPU architecture. Experimental results have shown that, the proposed implementation method achieved approximately 6 times faster results than implementation on CPU. But there is need to improve search performance in the steady-state model. There is also need to adopt different tasks that have non uniform computational granularity so that improved steady-state models will be more suited to the GPU environment [16].

An image matching algorithm based on genetic algorithm on GPU is implemented by Yongzhen Ke [17]. Designed fitness function, chromosomal selection function, crossover, and mutation function are suitable for running on GPU. The experimental results have proved that the proposed algorithm can match optimal result quickly and achieves satisfying speedup under the conditions of the existing hardware.

Peter Pospichal and Jiri Jaros[18] have implemented GA on GPGPU and achieved impressive speedup. Also, high quality solutions are obtained on nvidia GPU supporting Shader Model 4.0 and Linux/Windows platform. Innovative results are obtained and have proved that GA can achieve fast solutions to NP problems and requires less computational cost.

Petr Pospichal, Jiri Jaros, and Josef Schwarz [19] have showed that GPU results are 20% better than CPU. Also they proved that their proposed GPU implementation of GA is able to optimize numerical functions. They have concluded that they have obtained speedups up to seven thousand times while using GPU. This clearly showed that GPUs have proven their abilities for acceleration of genetic algorithms. During the optimization of simple numerical functions, GPU showed good performance and in a short time or in equal time. They have decided to put forward the future work to introduce more complex numerical optimization inspired by real-world problems. Moreover, they have decided to compare parallel island-based GA running on CPU with their proposed GPU version.

Noriyuki Fujimoto and Shigeyoshi Tsutsui [20] have showed the effect of parallelizing the OX operator on the NVIDIA CUDA GPU architecture. Their approach seems to be effective if no local search heuristic is known for the target problem. Their future work includes parallelizing other genetic operators in permutation domains.

Frédéric Pinel, Bernabé Dorronsoro and Pascal Bouvry [21] have extended their preliminary algorithm with a new design for a parallel implementation of Min–min algorithm both on a multi-core CPU and a GPU. The design they have proposed works fast and accurately for scheduling of independent tasks. Additionally, they have explored the two recombination operators designed for the GPU,

V. CONCLUSION AND FUTURE WORK

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

- ✓ GPGPU is good option for speedup to solve combinatorial problem.
- ✓ 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] B. Cheang a, H. Li b, A. Lim c, B. Rodrigues “Nurse rostering problems-a bibliographic survey”, European Journal of Operational Research 151, Elsevier-2004.

[2] 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

[3] 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.

[4] Cipriano, L. Di Gaspero, A. Dovier, “Hybrid approaches for Rostering 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.

[5] F. He, R. Qu, “A constraint programming based column generation approach to nurse rostering problems”, Computers & Operations Research 39 (2012) 3331–3343.

[6] 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

[7] R. Qu, F. He, “A hybrid constraint programming approach for nurse rostering problems ”, in: SGAI International Conference .

[8] 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.

[9] Swapnaja S. Balekar, Prof Nalini Mhetre, “survey of genetic algorithm approach for solving nurse scheduling problem.” IJSR 2015.

[10] Kathryn A. Uwe Aickelin “An Indirect Genetic Algorithm for a Nurse Scheduling Problem” Computer and Operatoons research,31(5),pp761778,2004

[11] Gareth R. Beddoe, Sanja Petrovic "Selecting and weighting features using a genetic algorithm in a case based reasoning approach to personnel rostering".

[12] Nourah Al-Angari , Abdullatif ALAbdullatif "Multiprocessor Scheduling Using Parallel Genetic Algorithm" 1,2 Computer Science Department, College of Computer & Information Sciences, King Saud University .

[13] Mohammed A. Awadallah , Ahamad Tajudin Khader , Mohammed Azmi Al-Betar 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications "Nurse Scheduling Using Harmony Search"

[14] Razamin Ramli1+ and Siti Nurin Ima Ahmad2 "Innovative Neighbor Generations In Tabu Search Technique" 2011 International Conference on Information and Electronics Engineering IPCSIT vol.6 (2011) © (2011) IACSIT Press, Singapore

[15] Edmund Burke and Peter Cowling "A Memetic Approach to the Nurse Rostering Problem" Applied Intelligence 15, 199–214, 2001

[16] Masashi Oiso, Yoshiyuki Matumura "Accelerating Steady-State Genetic Algorithms based on CUDA Archi tecture"

[17] Yongzhen Ke, Yuhao Li, Dandan Li "Image Matching using Genetic Algorithm on GPU"

[18] Peter Pospichal and Jiri Jaros[18] "GPU based acceleration of Genetic Algorithm."

[19] Petr Pospichal, Jiri Jaros, and Josef Schwarz "Parallel Genetic Algorithm on the CUDA Architecture"

[20] Noriyuki Fujimoto, Shigeyoshi Tsutsui "Parallelizing a Genetic Operator for GPUs" 2013 IEEE Congress on Evolutionary Computation June 20-23, Cancún, México

[21] Frederic Pinel a , Bernabe Dorronsoro b,∗, Pascal Bouvry "Solving very large instances of the scheduling of independent tasks problem on the GPU" J. Parallel Distrib. Comput. journal homepage: www.elsevier.com/ locate/jpdc.