Survey of Genetic Algorithm Approach for Nurse Scheduling Problem

Swapnaja S. Balekar¹, N. A. Mhetre²

¹ME Student, Department of CE, Smt. Kashibai Navale College of Engineering, Vadgaon, MS, India.
²Assistant 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.
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 NRP 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 MGGG(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


[38] Jau-Ming Su, Jen-Yu Huang, “Using Ant Colony Optimization to Solve Train Timetabling Problem of Mass Rapid Transit”


