Genetic Algorithm for Scheduling in a Grid

Namrata Mahakalkar
Department of Computer Science and Engineering
P.I.E.T, Nagpur,

A.R.Mahajan
Department of Computer Science and Engineering
P.I.E.T, Nagpur

ABSTRACT

The computational grid provides a promising platform for the deployment of various high-performance computing applications. In computational grid, an efficient scheduling of task onto the processors that minimizes the entire execution time is vital for achieving a high performance.

High throughput computing (HTC) is of great importance in grid computing environments. HTC is aimed at minimizing the total makespan of all of the tasks submitted to the grid environment in long execution of the system. To achieve HTC in grids, suitable task scheduling algorithms should be applied to dispatch the submitted tasks to the computational resources appropriately.

In this paper we present a Genetic Algorithm approach for scheduling operating room (OR) nurses. Most studies in operating room scheduling deal with patient flow analysis and physician scheduling, limited literature has focused on scheduling OR nurses. Our objective is to minimize nurses' idle time, overtime and non-consecutive assignments during overtime hours while maximizing demand satisfaction. The major constraints are: 1) shift constraints and 2) match between nurses' skill sets and surgery requirements. Due to the large size of the problem, finding an optimal solution is extremely difficult. Therefore, a Genetic Algorithms approach is proposed to find a set of good schedules in a reasonable amount of time.

Keywords

Genetic algorithm, Grid computing

1. INTRODUCTION

The popularity of the Internet and the availability of powerful computers and high-speed networks as low-cost commodity components are changing the way we use computers today [1]. These technical opportunities have led to the possibility of using geographically distributed and multi-owner resources to solve large-scale problems in science, engineering, and commerce [1].Recent research on these topics has led to the emergence of a new paradigm known as grid computing [2].The emerging paradigm of grid computing and the construction of computational grids [3] are making the development of large scale applications possible from optimization and other fields. The development or adaptation of applications for grid environments is being challenged by the need of scheduling a large number of jobs to resources efficiently. Moreover, the computing resources may vary in regard to their objectives, scope, structure as well as to their resource management policies such as access and cost[12]. Grid resource management provides functionality for discovery and publishing of resources as well as scheduling, submission and monitoring of jobs. Most functions are standardized in the Grid middleware Globus toolkit. One exception is job scheduling, a key element in resource management that needs more sophisticated implementations because of the heterogeneity of the Grid nodes. One interesting implementation method is to build job scheduling services on top of Grid middleware to meet the requirements of specific applications [14]. Effective job scheduling in Grid requires to model the available resources on Grid nodes and computation requests of jobs, determine the current load of the system, and predict the job execution time. Job scheduling in parallel and cluster computing is focussed on estimating the system load from experience with a performance model [15]. Their goals are to achieve best performance and load balancing across the entire system. When applied to Grid environments those methods often result in poor performance due to the heterogeneity of Grid resources. Grid job scheduling focuses on the job execution time because of the open, dynamic Grid environment [16].

2. NAP

The nurse assignment problem in the operating suite is very difficult to solve and large problem instances often result in unreasonably long solution times. In a given day, in the operating suite, there may be up to 70 cases and 150 nurses. It can be difficult to find a good schedule in a reasonable amount of time. A schedule should specify that a given nurse is assigned to which OR at what time and what job is to be performed by the nurse.

Based upon the deterministic nurse assignment optimization model developed the Nurse Assignment Model (NAM) described in this paper is a multi-objective optimization model. The objectives are minimizing demand dissatisfaction, nurse idle time and overtime. The available nurses are assigned to surgery cases that are scheduled for that day in the surgical suite. This assignment is based on the requirements of the surgery case such as specialty, complexity, number of registered nurses (RNs) and scrub technicians (Techs) that are needed for the surgery cases that are scheduled to be performed in a day. The duration of the surgery is estimated by the surgeons.

We make the following assumptions to develop the NAM. Working days are divided to 30-minute time intervals. There are eight-hour, ten-hour and twelve-hour regular shifts that include regular hours and authorized overtime hours. We assume that when a nurse is assigned to a case, he/she stays there till the case is done. A nurse with a higher competency level can also be assigned to less complex surgeries. Minimum requirements for surgeries are considered. That means only one RN and one Scrub Tech are assigned to each surgery case. This assumption is not misleading because this is the worst case and if there is any nurse remaining the nurse manager can assign extra nurses if it is needed. The decision variable is a three dimensional binary variable which has the value of one, if nurse *i* is assigned to case *c* to do role *k*. The Objective is to minimize total idle time, total overtime and demand dissatisfaction. Tables 1 and 2 show the notation used in the optimization model for sets and parameters respectively.

Table 1. Notation of sets used in Nurse Assignment Model

Notation	Description
I	Set of available nurses
K	Set of roles that are required for each surgery case (k=1 RN, k=2 Tech)
Q	Set of specialties
C	Set of cases scheduled for surgery in each day
S	Set of available shifts
P	Set of competency/complexity levels (1: Simple, 2: Moderate, 3: Complex)
H	Time intervals in a working day

2.1 System Definition

The system under study is an operating suite with 31 operating rooms each dedicated for surgeries of certain types. The rooms are equipped according to the requirements of surgeries that are performed in them.

The surgeries are categorized into 17 main medical specialties, such as Urology, Cardio, Neuro, etc. Within each medical type of surgery there is another classification: a surgery may be complex, moderate or simple. Each surgery case demands at least one of each type of nurses, one Registered Nurse (RN) and one Scrub Technician (Scrub tech). The specialty and competency level of the nurses that are assigned to a surgery case must match the surgery specialty and complexity.

The duration of cases varies from one type of surgery to another and also depends on the complexity of the case. The surgery cases are scheduled and assigned to ORs based on the availability of ORs and the estimated duration of the surgery which is determined by the surgeon. However, the surgery may take longer or shorter than the estimated duration. Comparing observed case duration data with the parametrically estimated values shows that, in general, the estimates can be far from the real value for the surgery duration. Of course, a poorly designed schedule due to inaccurate estimates will lead to underutilization or overutilization of staff and overtime of the OR suite. In this study we try to include the stochastic nature of case durations in assignment of nurses to operating rooms.

The operating suite has an average of 150 nurses, having 60 nurses available in any given day. There are two types of nurses RNs and Scrub Technicians. Some of the RNs have the skill of doing scrub but the opposite is not true. Nurses also have medical specialties. Each nurse can work on a surgery case that matches his/her specialty. Within each specialty, nurses also have different competency levels that means, based on their skills and experience, they can work on a simple, moderate or complex surgery case. The ORs start working at 6:30 a.m. and officially finishes at 11:00 p.m. There are five different shift patterns in a day, some of them are eight-hour and others are ten-hour shifts with different starting times.

2.2 Scheduling and rostering

According to the research, the terms scheduling and rostering are defined as follows:-
1) **Scheduling** is the allocation, subject to constraints, of resources to objects begins placed in space-time, in such a way as to minimise the total cost of some set of the resources used.
2) **Rostering** is the placing, subject to constraints, of resources into slots in a pattern.
solution. Result will show the Schedule for Staff for the given number of shifts in a planning period by considering constraints. We represent the model by using GUI in MATLAB.

2.3 Constraints

This section describes the hard and soft constraints for nurse scheduling problem described by Dean, Adamuthe et.al. Uwe Aickelin et.al. [11][12][13]

2.3.1 Hard Constraints

Hard constraints are those that must be satisfied. Violation of these constraints (also called as conflicts) will cause the solution to be infeasible which is not accepted. Solutions which satisfy hard constraints are called feasible solutions.
HC1: Create a 4-week schedule. The hard constraint varies from defied by Dean. Dean has proposed solution by considering Sunday as start day where as in this paper we have considered Monday is start day.
HC2: Fixed number of working employees as required during each day of planning period.
HC3: Each employee works at least five days per week.
HC4: Maximum number of consecutive working days for any employee is six.
HC5: Minimum time gap required between two shifts should be of 12 hours.

2.3.2 Soft Constraints

Soft constraints are those that are desirable in order to produce a good quality timetable but violations are allowed to satisfy hard constraints.
Cost Minimization Constraints:
SC1: Try to avoid more than 3 working days per employee per week because that leads to overtime pay.
SC2: Try hard to avoid more than 4 working days per employee per week because that leads to additional overtime pay.
Personal Demands:
SC3: If an individual wants to avoid working 3 days in a row try to accommodate.
SC4: If an individual wants to maximize the number of grouped working days, then try to maximize the number of grouped working days.
General:
SC5: Try to balance the number of Monday and Friday off days that coincide with off weekends.

In this paper, focus would be to solve the hard constraints only as making them the primary priority for nurse scheduling. Such that the main objective for solving the problem would be allocation of nurses or staff to particular slot in planning period.

3. DESCRIPTION OF METHODOLOGY

3.1. Background of Genetic Algorithm

Goldberg describes Genetic Algorithms as: search procedures based on the mechanics of natural selection and natural genetics. I.e. they are general search and optimisation algorithms that use the theories of evolution as a tool to solve problems in science and engineering. This involves evolving a population of candidate solutions to the particular problem, using operations inspired by natural genetic variation and natural selection.

Genetic Algorithms are 'weak' optimisation methods. That is they do not use domain-specific knowledge in their search procedure. For this reason they can be used to solve a wide range of problems. The disadvantage, of course, is that they may not perform as well as algorithms designed specifically to solve a given problem.

From the very beginning, computer scientists have thought of systems that would mimic one or more attributes of life. However, it wasn't until the 1960s that Genetic Algorithms (GAs) were formally developed by John Holland, along with his students and colleagues from the University of Michigan. Holland's original goal, however, was not to design algorithms to solve specific problems, as in other evolutionary programming, but to study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems.

Holland's GA is a method of moving from one population of chromosomes (strings of ones and zeros) to a new population using a kind of natural selection, together with the genetics-inspired operations of crossover, mutation and inversion,

A typical algorithm might consist of the following:

A number of randomly chosen guesses of the solution to the problem - the initial population.

- ❖ A means of calculating how good or bad each guess is within the population - a population fitness function.
- ❖ A method for mixing fragments of the better solutions to form new and on average even better solutions - crossover.
- ❖ An operator to avoid permanent loss of (and to introduce new) diversity within the solutions – mutation.

With these being the basic components of most GAs it can be seen that they are a simple method to solve a specific problem. The downside, however, is that there are many different ways of performing these steps. In this dissertation I have attempted to provide a package that gives the user a choice of using some of the more common methods to solve their particular problem.

3.2 Proposed Approach for NSP based on Genetic Algorithm

This paper will use real number coding such that the data would be presented in real number. The chromosomes of GA are a long string with 840 elements, which stores the information of all the employees. Since each nurse is assigned to one and only one shift per day, each nurse has seven elements for four weeks and each elements represents for the type of shifts for that day.
The integer numbers used for representing different types of shifts are given below:-
Free shift- integer 0
Shift 1- integer 1
Shift 2- integer 2
Shift 3- integer 3
Shift 4- integer 4
Shift 5- integer 5
As there are 30 nurses in this problem, the chromosomes would include 840 elements (30*7*4)
30-represents the number of nurses, 7-no.of days and 4-no. of week needed to be scheduled.

3.3 Parameters used while encoding for GA

For solving specific type of problems in genetic algorithm encoding is required. Different parameters are needed for the purpose of encoding; some parameters are listed below in the table used for solving NSP,

Parameter/Strategy	Setting
Elite count	1
No.of initial population	100
Length of Chromosome	840
Crossover rate	0.8
Mutation rate	0.002
Max no of generation	500
Stall Gereration	100

Designing of GUI for NSP

This window gives the main appearance for the nurse scheduling problem where different parameters were set on the left side using toggle button and the blank side gives the result whenever the button is clicked.

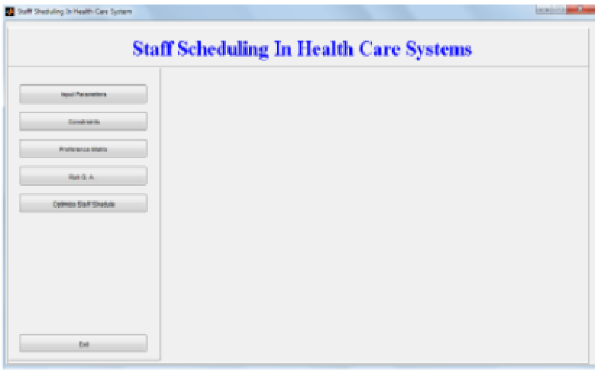

Fig1: GUI Window for NSP

Fig 2: GUI Windows when input parameter button is clicked

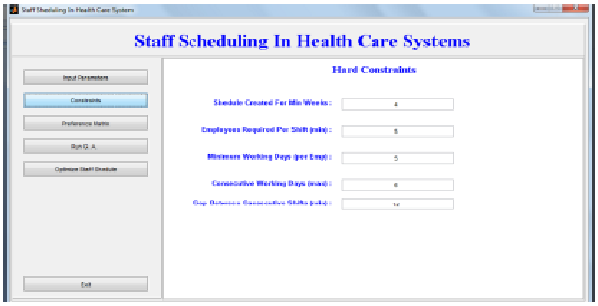

Fig 3: GUI Windows when constraints button is clicked

Similarly, different buttons are clicked and the output is viewed in the GUI window according to the requirements of the user. GUI helps in reducing the time involved for solving mathematical expressions and gives a perfect pattern for presenting the problem along with the solution.

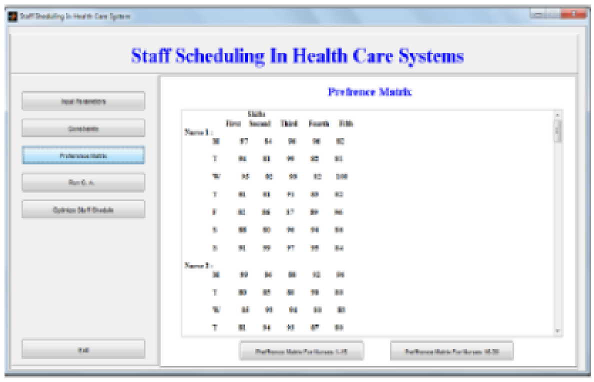

Fig 4: GUI windows when preference matrix button is clicked

4. CONCLUSIONS

The above problem will now be applied in a Grid or distributed environment. We are not getting an optimized solution for a serial GA. We will be applying parallel GA to the particular problem and in a Grid environment in the future.

5. REFERENCES

[1] Fangpeng Dong and Selim G. Akl, "Scheduling Algorithms for Grid Computing": State of the Art and Open Problems, School of Computing, Queen's University Kingston, Ontario January 2006.

[2] R. Buyya and D. Abramson and J. Giddy and H. Stockinger,"Economic Models for Resource Management and Scheduling in Grid Computing", in J. of Concurrency and Computation: Practice and Experience, Volume 14, Issue.13- 15, pp. 1507-1542,Wiley Press,

[3] Goux, J. P. and Leyffer, S. Solving large MINLPs on computational grids. *Optimization and Engineering*, Vol 3, 2002, p. 327.346.

[4] E. Alba, A.J. Nebro, J.M. Troya, ”Heterogeneous Computing and Parallel Genetic Algorithms,” Journal of Parallel and Distributed Computing 62, pp. 1362-1385, 2002. 21

[5] E. Alba, J.M. Troya, ”Synchronous and Asynchronous Parallel Distributed Genetic Algorithms,” Future Generation Computer Systems, 17(4):451-465, January 2001.

[6] M. Nowostawski, R. Poli, ”Parallel Genetic Algorithm Taxonomy,” Proceedings of the Third International conference on knowledge-based intelligent information engineering systems (KES’99), pp. 88-92, Adelaide, 1999.

[7] E. Cantu-Paz, ”A Survey of Parallel Genetic Algorithms,” Calculateurs Paralleles, Reseaux et Systems Repartis vol. 10 No. 2 pp. 141-171, 1998.

[8] F.J. Villegas, T. Cwik, Y. Rahmat-Samii, and M. Manteghi, ”A parallel electromagnetic Generic-Algorithm Optimization (EGO) application for patch antenna design,” IEEE Transactions on Antennas and Propagation, Vol. 52, No. 9, pp. 2424-2435, 2004.

[9] D. Abramson, J. Abela, ”A Parallel Genetic Algorithm for Solving the School Timetabling Problem,” Technical Report, Division of Information Technology, C.S.I.R.O, Melbourne, 1991.

[10] Edmund K. Burke et al. *The State of The Art of Nurse Rostering* Journal of Scheduling *7: 441–499, 2004.* ©*2004* Kluwer Academic Publishers. Printed in the Netherlands.

[11] Tiago M. Dias, Daniel F. Ferber, Cid C. de Souza and Arnaldo V. Moura, *Constructing nurse schedules at large hospital* International Transaction in Operational Research *10 (2003) 245–265*

[12] Amol C. Adamuthe and Rajankumar Bichkar, *Hybrid Genetic Algorithmic Approaches for Personnel Timetabling and Scheduling Problems in Healthcare* International Conference on Technology Systems and Management (*ICTSM) 2011* Proceedings published by International Journal of Computer Applications® (IJCA)

[13] Ivo Blochliger, *Modelling sta*ff *scheduling problems* A tutorial European Journal of Operational Research *158 (2004) 533–542*

[14] K. Krauter, R. Buyya, M. Maheswaran,Ataxonomy and survey of Grid resource management systems for distributed computing, Software Pract. Exp. 2 (2002) 135–164.

[15] Y. Zhang, H. Ranke, J.E. Moreira, A. Sivasubramaniam, An integrated approach to parallel scheduling using gang-scheduling, backfilling, and migration, in: Lecture Notes in Computer Science, Vol. 2221, Springer, Berlin, 2001, pp. 133–158.

[16] D. Thain, T. Tannenbaum, M. Livny, Condor and the Grid, in: A.J.G. Hey, F. Berman, G.C. Fox (Eds.), Grid Computering: Making the Global Infrastructure a Reality,Wiley,West Sussex, England, 2003, pp. 299–335.