IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 10, October 2015.
www.ijiset.com
ISSN 2348 – 7968

A Web Based Framework for Solving Timetabling Problem Using Parallel Genetic Algorithm with Local Search

Milena Lazarova[1], Neli Lepoeva[2]

[1,2] Department Computer Systems, Technical University of Sofia
Sofia, Bulgaria

Abstract

The paper presents a web based framework that provides possibility for efficient solving of a timetabling problem using parallel genetic algorithm with local search. The parallel computation model is based on a coarse-grained distributed genetic algorithm with island based local evolution of subpopulations and circular migration. A hybrid approach that uses a local search applied for each individual in the population further intensifies the optimization process. The web based framework provides means to set up particular timetabling problem modifying the required input data as well as the hard and soft constraints. It also ensures remote execution of the parallel genetic optimization and gathers certain statistics of several runs. Based on that statistics the optimization parameters are analyzed and tuned. The framework also gives tools for verification and modification of the output results for direct usage in given educational institution.

Keywords: *Timetabling problem, Local search, Genetic algorithm, Parallel model, Web application.*

1. Introduction

The timetabling problem is considered as the task to create a schedule and is applied in many different modifications such as timetabling in educational institutions, sport events, conference programs, etc. It is an optimization problem aimed at finding a solution that meets given set of constraints and thus optimizes certain objective function [1, 2]. The timetabling problem can be solved using different approaches: search strategies as backtracking or constraint logic programming [3, 4]. Since the construction of an optimal time schedule is a combinatorial problem of NP-complexity, a metaheuristics approaches can be efficiently applied.

The interest in application of different metaheuristics or combinations of them for solving hard optimization problems grows rapidly in the last decade. There is also an intensive interest in the parallelization of the metaheuristics, first because of most of the metaheuristics require execution of computationally intensive algorithms that find sub-optimal solution of hard combinatorial optimization problems, and second due to the fast developments in the field of parallel computer platforms and widespread application of parallel programming models and implementations.

The paper presents a web based framework that provides possibility for efficient solving of a timetabling problem using parallel genetic algorithm with local search. The parallel computation model is based on a coarse-grained distributed genetic algorithm with island based local evolution of subpopulations and circular migration. A hybrid approach that uses a local search applied for each individual in the population further intensifies the optimization process. The web based framework provides means to set up particular timetabling problem modifying the required input data as well as the hard and soft constraints. It also ensures remote execution of the parallel genetic optimization and gathers certain statistics of several runs. Based on that statistics the optimization parameters are analyzed and tuned. The framework also gives tools for verification and modification of the output results for direct usage in given educational institution.

2. Solving a timetabling problem using genetic algorithm

2.1 Timetabling problem

The timetabling problem is usually defined as allocation of numerous events (lessons, lectures, seminars, etc.) to a number of time intervals so that certain predefined constraints to be satisfied. The constraints impose some restrictions and can be regarded as hard and soft. The hard constraints are those which should not be violated in the suggested timetabling solution, examples are a student cannot attend two different lessons at the same time, a teacher cannot present two different lectures at the same time. Soft restrictions are considered as preferences that might or might not be satisfied, for example students should not attend more than three consecutive sessions.

ISSN 2348 – 7968

The problem input data comprises multiple events or classes: $E = \{e_1, ..., e_n\}$, which can be set to a set of time intervals (5 days, 9 timeslots per day): $T = \{t_1, ..., t_{45}\}$, a set of rooms for the events: $R = \{r_1, ..., r_m\}$, a set of student groups that attend classes: $S = \{s_1, ..., s_p\}$ and possibly a required equipment which activities require: $F = \{f_1, ..., f_q\}$, Every room $r_k \in R$ has many equipment $f_k \subseteq F$, while each event requires certain equipment $f_i \subseteq F$. Each student group should attends given set of classes and each room has a capacity of persons it can hold.

A solution to the timetabling problem is such distribution of students groups to have classes at given timeslots and rooms, so that a set of restrictions to be satisfied. The requirements to the schedule might differ from institution to institution and are given as a list of constraints, for example:

- one lecturer can only give one lesson to one student group in one room at the same time;
- one student group can only attend one lesson in one room at the same time;
- the assigned rooms are large enough to accommodate the student group attending the scheduled class;
- the required equipment for certain class is available at the assigned room.

A penalty value is added to the solution objective function if any of the soft constraints is violated:

- more than predefined maximum number of lessons per day are assigned for a lecturer;
- more than predefined maximum number of lessons per day are assigned for a student group;
- time periods free of lessons are assigned for a lecturer;
- time periods free of lessons are assigned for a student group;
- a student group has only one class per day.

Some of the soft constraints can be verified without taking into account the overall time schedule, while others can only be checked after the solution is fully constructed and all events are assigned to time intervals. The objective function of the problem minimizes the number of violations of the soft restrictions while only considers valid solution, i.e. those with hard constraints fully satisfied.

Increasing the size of the input data (student groups, lecturers, events, rooms) the number of all possible solution grows exponentially. Appropriate approach to find a solution is to introduce a metaheuristics method either based on a solution space search, population based or a hybrid heuristics.

2.2 Genetic algorithm

Genetic algorithms (GAs) are computational approaches for solving a variety of optimization problems [5, 6]. GAs are search procedures based on the ideas of evolutionary processes in the biological individuals. The general sequence of steps for solving the optimization problem by a genetic algorithm is illustrated on fig.1.

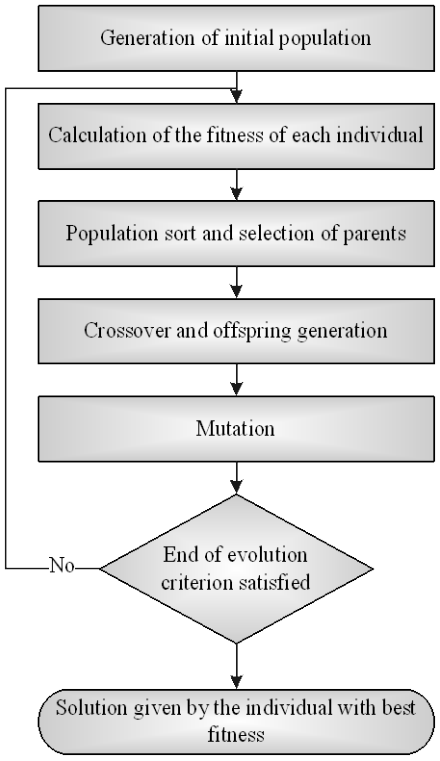

Fig. 1 Basic steps of a genetic algorithm.

Initial population of individuals (solutions) is usually generated randomly providing given number of possible solutions to the problem and the objective function of each solution is evaluated and considered as individual's fitness. At the selection step the solutions with higher fitness value have higher chance of being selected for reproduction stage. At the reproduction stage a new solutions are generated based on the selected parent solutions: the crossover combines the parent solution to create a new individual and the mutation applies minor changes to selected individuals.

2.3 Solving timetabling problem with genetic algorithm

Different metaheuristics can be applied for solving the timetabling problem [7, 8, 9]. In the paper a memetic

ISSN 2348 – 7968

algorithm based on a combination of a genetic algorithm and a local search is used. In order to apply a genetic algorithms for solving the timetabling problem the solution is represented as a list of ordered pairs (time slot, room) of size equal to the number of the events. The pair position corresponds to the event number. The room distribution is not part of the explicit solution and a max matching algorithm is used that purpose. For each time slot and the assigned events a list of rooms that satisfies the requirements of the events is generated. Max matching connects the events to the rooms using deterministic network flow algorithm.

A representation of the time schedule in this manner allows easy definition of a neighborhood of given solution generated only by movements of time periods and events. The neighborhood N is regarded as a union of two smaller neighborhoods: N_1 defined by movement operator that changes the time slot of an event and N_2 defined by a change operator that swaps the time slots of two events. In order a solution to be valid all hard constrains must be satisfied. Thus a stochastic local search of the above described neighborhood N is applied for the offspring generated by the genetic algorithm. The local search also generates modified solutions as an attempt to satisfy the soft constraints.

In order to speed up and to improve the efficiency of the described local search approach it is applied in a hybrid metaheuristics in combination with a genetic algorithm. The genetic algorithm uses heuristic information of the local search as described in [10]:

- evolution of steady generations: (i) only one pair of parents is selected for reproduction; (ii) the generated offspring replaces the worst fitted individual in the generation;
- selection follows a random selection approach;
- fitness function depends on the number of violations of the hard and the soft restrictions;
- crossover is based on equal probability of inheritance of the time slot assignment of each event from either of the two parents;
- mutation consists of an arbitrary move in the neighborhood space as defined by the local search approach.

The memetic algorithm uses a local search applied for each individual in the population. Important aspect of that hybrid approach is the balance between the number of steps of the local search for efficient search space exploration and the number of generations of the evolutionary algorithm for search space diversification.

3. Parallel genetic algorithm with local search

The strategies for parallel computations of the population based metaheuristics can either utilize concurrent computations of the operation of each individuals thus decreasing the overall computational time for finding a solution, or parallel evolution approach that employs population split and concurrent evolution of subpopulations.

The models of parallel genetic algorithms (PGA) can be categorized as:

- standard models where the whole population is a common pool of individuals;
- structured models utilizing decentralized population.

The population pool models are further discriminated depending on the reproduction stage providing two algorithmic classes with different granularity: generation based models that completely replace the parents' individuals by the offspring, and stable state models that generate new individuals joining the population. The decentralized population model uses migration between islands of the subpopulation evolution (distributed GA) or pollination in a diffuse lattice (cell GA). The parallel computational models for GA also depend on the topology of the migration/diffusion (circular or global, one or bidirectional) and the mutation strategy (variable mutation rate, self-adaptive mutation of the individuals) [11].

The PGA model used in the framework described in this paper is based on a coarse-grained distributed genetic algorithm with a data parallelism. Parallel computational paradigm SPMD (Single Program Multiple Data) with synchronous iterations and periodic two-way circular migration is utilized (fig. 2).

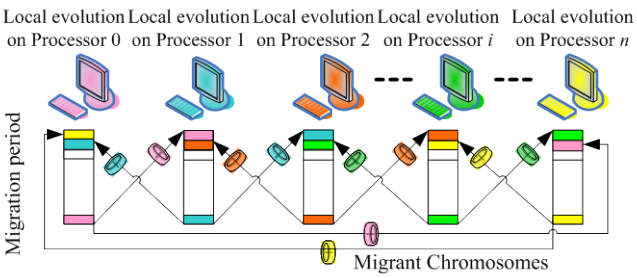

Fig.2 Parallel genetic model with periodic two-way circular migration

ISSN 2348 – 7968

Each population evolves locally within a single process (island). In order to improve the convergence of the PGA to solutions of high quality, a periodic migration of individuals between the islands occur: the most fitted individuals of each local evolution are transferred to other local islands replacing the weakest local individuals in the given subpopulation. The efficiency of the PGA with circular migration depends on a set of parameters including initial size of the subpopulations, the intensity of migration, the number of migrants. In the presented approach a topology of the virtual migration channels is applied that ensures effective exchange of the best individuals between islands (fig. 3).

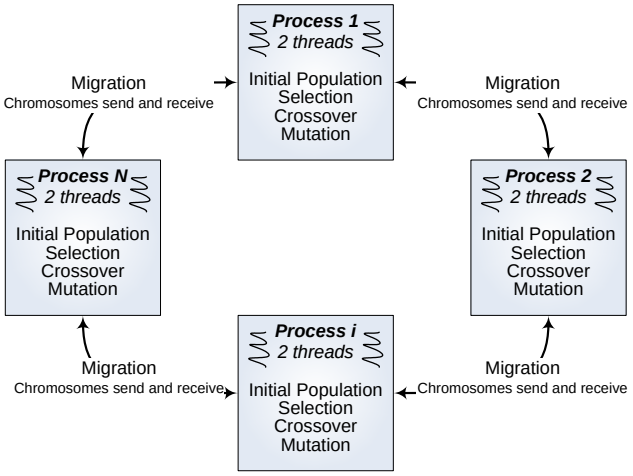

Fig.3 Parallel computational model of GA with local search

4. Web based framework for solving timetabling problem using PGA with local search

The main goal of the developed web based application for solving timetabling problem using parallel genetic algorithm with local search is to provide an interface for input and output data management, remote parallel execution of the genetic algorithm, gathering statistics and analyses of the results. The framework is based on a client-server architecture with presentation, application and database access layers (fig. 4).

The client part represents a single page application which provides menu navigation and user interface to the framework. The client user interface is implemented using Polymer library [15]. The two-way communication in real time between the client and the server is based on web sockets which are part of HTML5.

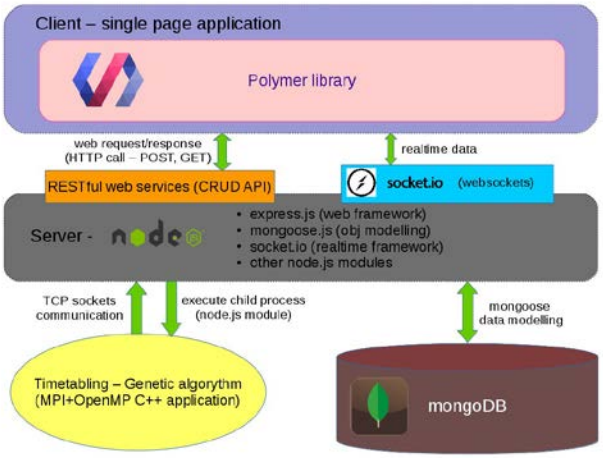

Fig.4 Architecture of the web based framework for solving timetabling problem using PGA with local search

The server part is developed using node.js [16]. Being a platform independent it allows direct implementation of JavaScript to the server thus providing a means to create fast scalable web applications. RESTful web services are used through express.js module to process web requests and responses. Besides node.js uses an event driven model featuring non-blocking input and output operations. This is an advantage and makes it efficient choice for data real time applications using large volume of distributed data.

The database used in the framework is MongoDB. Being a non-relational database it provides use of a distributed storage with good scalability. Data storage required for the web framework are organized in MongoDB as NoSQL collections that stores structured information in JSON format with dynamic schemes. This makes data integration easier and faster.

For remote execution of the optimization process based on the parallel genetic algorithm with local search a child process of node.js module is executed. Thus the PGA is launched as a separate process that uses one way communication with the server by JSON messages over TCP sockets.

The server part uses mongoose module of node.js in order to connect the database, store and retrieve data. On the other side the client part uses HTTP request to the server to retrieve and visualize results and statistics of the optimization.

The parallel computational model of the genetic algorithm is implemented in C ++ using the MPI and OpenMP.

ISSN 2348 – 7968

The user interface of the client part of the framework has the following functionality:

- execution of PGA for timetabling (fig. 5, fig. 6):
  - set input data for events, students groups, rooms;
  - set hard and soft limits;
  - set local search and genetic algorithm parameters;
  - set parallel execution parameters;
- output results visualization (fig. 7, fig. 8):
  - real time visualization of log data of the PGA execution;
  - preview and modification of the generated time schedule;
  - visualization of output results from PGA execution stored in the database: by solution quality, by execution time, by input data file, by local search parameters, by genetic algorithm parameters, by parallel execution parameters;
- statistics visualization (fig. 9, fig. 10):
  - visualization of charts for solution quality and parallel performance parameters.

The developed web based framework is tested using a heterogeneous parallel cluster with the following parameters: CPU AMD Opteron 64 Dual Core Processor 1.8 GHz, RAM 2GB 800 MHz, HDD: 2x160GB Hitachi SATA in RAID 0, MB Asus M2N-LR, OS Scientific Linux 6.3 64 bit, Middleware SLURM, GLUSTERFS, MPICH2 1.4.1p2.

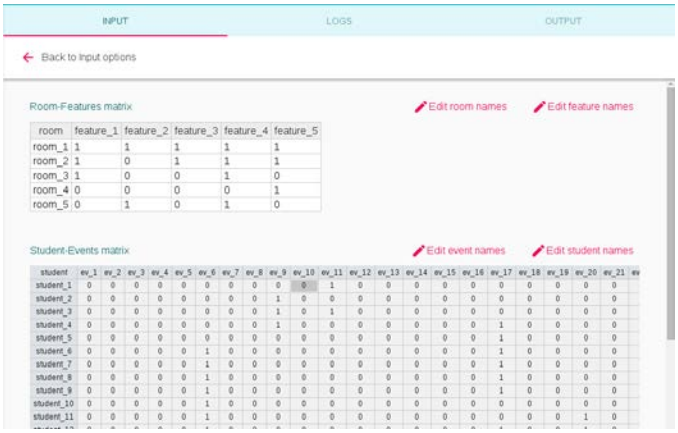

Fig.5 Web framework interface for input data modification

Fig.6 Web framework interface for local search parameters

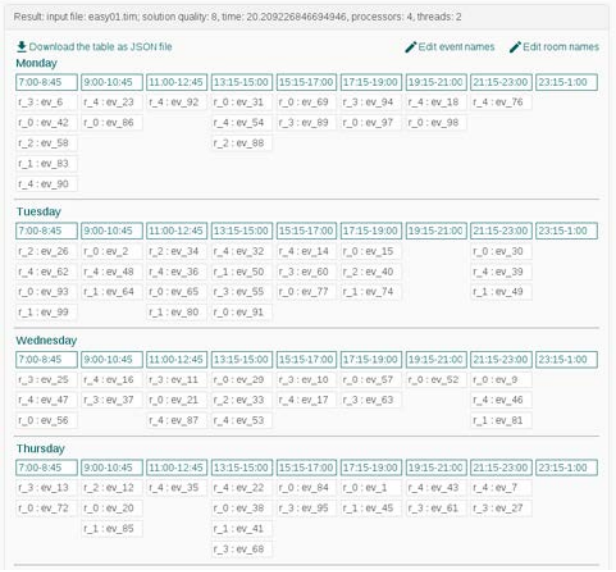

Fig.7 Visualization of the timetabling solution

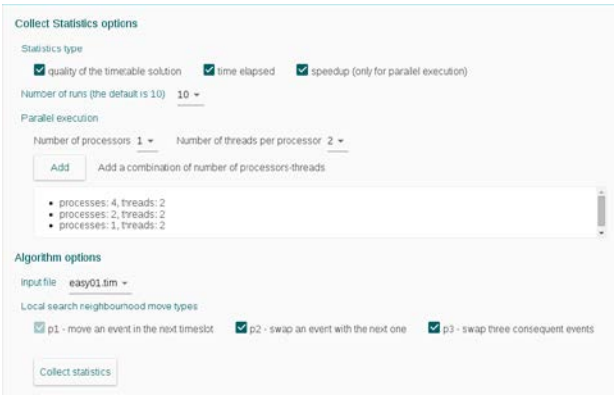

Fig.8 Visualization of the timetabling statistics options

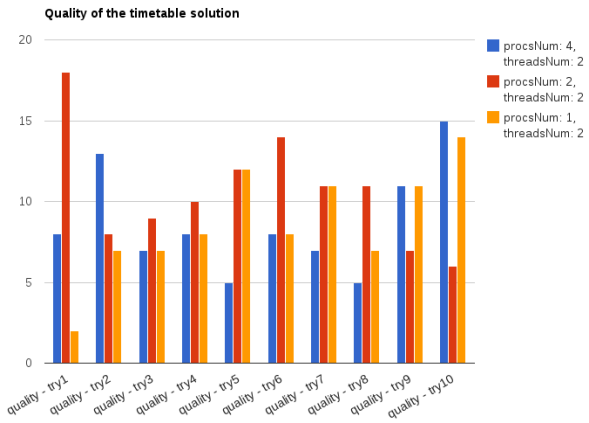

Fig.9 Visualization of statistics for quality of timetable solution

ISSN 2348 – 7968

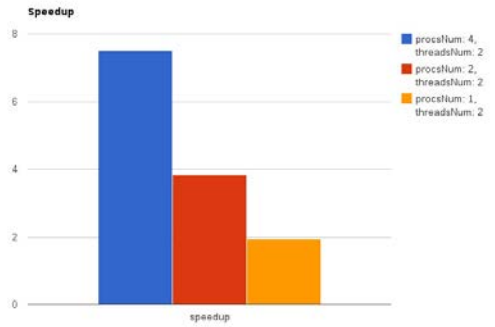

Fig.10 Visualization of statistics for speedup of parallel execution

4. Conclusions

The developed web based framework provide possibility for efficient solving of a timetabling problem using parallel genetic algorithm with local search. The input data are easily set and modified. The timetabling problem is solved using hybrid metaheuristics utilizing genetic algorithm and local search applied for each individual in the population. A parallel computation model based on distributed genetic algorithm with island based local evolution and circular migration is remotely executed. The optimization results as well as statistics of multiple independent runs of the memetic algorithm are gathered and stored in a non-relational database for better scalability. The framework also gives tools for verification and modification of the output results for direct usage in given educational institution.

References

[1] A. Schaerf, “A Survey of Automated Timetabling”, Artificial Intelligence Review, Vol. 13, 1999, pp. 87÷127.
[2] M. Grobner, P. Wilke, S. Buttcher, “A Standard Framework for Timetabling Problems”, Proc. of 4th Int. Conf. PATAT 2002, Gent, Belgium, Aug. 2002, LNCS 2740, pp. 24÷38.
[3] S. A. MirHassani, F. Habibi, “Solution approaches to the course timetabling problem”, Artificial Intelligence Review, Feb. 2013, Vol. 39, No. 2, pp. 133÷149.
[4] H. Rudova, T. Müller, K. Murray, “Complex university course timetabling”, Journal of Scheduling April 2011, Vol. 14, No. 2, pp.187÷207.
[5] A. Eiben, “Evolutionary Algorithms and Constraints Satisfaction: Definitions, Survey, Methodology, and Research Directions”, in L. Kallel, B. Naudts, A. Rogers (eds.), Theoretical Aspects of Evolutionary Computing, Natural Computing Series, Springer, pp.13÷58, 2001.
[6] C. Reeves, J. Rowe, Genetic Algorithms – Principles and Perspectives: A Guide to GA Theory, Springer, 2002.
[7] H. Babaeia, J. Karimpourb, A. Hadidic, “A survey of approaches for university course timetabling problem”, Journal Computers & Industrial Engineering, Vol. 86, Aug. 2015, pp. 43÷59.
[8] N. Pillay, “A survey of school timetabling research”, Annals of Operations Research, July 2014, Vol. 218, No. 1, pp. 261÷293.
[9] O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. M. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete and T. Stützle, “A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem”, Proc. of 4th Int. Conf. PATAT 2002, Gent, Belgium, Aug. 2002, pp. 329÷351.
[10] O. Rossi-Doria, C. Blum, J. Knowles, M. Sampels, K. Socha, B. Paechter, “A local search for the timetabling problem”, Proc. of 4th Int. Conf. PATAT 2002, Gent, Belgium, Aug. 2002, pp. 124÷127.
[11] E. Cantu-Paz, “Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms”, Journal of Heuristics, Vol. 7, No. 4, 2001, pp. 311÷334.
[12] S. Jat, S. Yang, “A Guided Search Genetic Algorithm for the University Course Timetabling Problem”, Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2009), Aug. 2009, Dublin, Ireland, pp.180÷191.
[13] P. Wilke, M. Groebner, N. Oster, “A Hybrid Genetic Algorithm for School Timetabling”, Advances in Artificial Intelligence, Vol. LNAI2557, Canberra, Australia, 2002. pp. 455÷464.
[14] B. Wilkinson, M. Allen, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers, Pearson Prentice Hall, 2005.
[15] www.polymer-project.org
[16] nodejs.org
[17] expressjs.com
[18] socket.io
[19] mongodb.org
[20] mongoosejs.com

Milena Lazarova, Assoc. Prof., PhD, MSc. Eng., has graduated from the Technical University of Sofia, Faculty Computer Systems and Control, specialty Computer Technologies in 1995. She is working in the Department "Computer Systems" at the Technical University of Sofia. Her research interests are in the field of parallel information processing, parallel algorithms, parallel programming, metaheuristics, image processing and analyses, computer graphics, pattern recognition.

Neli Lepoeva, MSc. Eng., Technical University of Sofia, Faculty Computer Systems and Control, specialty Computer ans Software Engineering in 2015. Her research interests are in the field of parallel algorithms, parallel programming, genetic algorithms and metaheuristics.