ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

Parallel Artificial Bee Colony Optimisation for Solving Curricula Time-Tabling Problem

Milena Lazarova

Associate Professor, Dept. of Computer Systems, Technical University of Sofia, Bulgaria

ABSTRACT: The paper presents parallel computational model for solving the curricula time tabling problem using hybrid metaheuristics approach that combines population based artificial bee colony optimization and trajectory based simulated annealing for local optimization. The suggested solution is targeted for multicomputer high performance architecture and exploits both shared memory and distributed memory parallel computational models utilizing fine grained thread level parallelism and master-slave message passing flat model. The experimental evaluation shows good scalability of the solution quality due to better diversification and local exploration of the search space when increasing the number of processes and threads. The speedup of the parallel computational model also scales almost linearly in respect to both the parallel workload and the machine size. In addition a web based application is developed to ease the schedule construction, editing, visualization and usage in educational institutions.

KEYWORDS: time-tabling problem; artificial bee colony; parallel computational model; shared memory; distributed memory; simulated annealing, web based application.

I. INTRODUCTION

The timetabling problem is considered as a task to create a schedule and is applied in many different modifications such astimetabling in educational institutions, sport events, conference programs, etc. The task to generate a schedule is considered to be hard optimisation problem.

An optimization problem consists in finding the best (cheapest, heaviest, etc.) element in a large set and usually specified implicitly, where the quality of elements of the set are evaluated using an objective function [1]. Combinatorial optimization problems have a discrete search space of the possible solutions. Many combinatorial optimization problems have high computational complexity and are classified as NP-hard problems. Heuristic techniques for solving optimization problems in an approximate way are aimed at finding near-optimal (or approximate) solution in reasonable time.

A metaheuristic is a general framework for heuristics in solving hard computational problems [2]. Metaheuristics are high level strategies for exploring search spaces by using different methods providing dynamic balance between diversification and intensification of the searched space [3]. The successful implementation of a metaheuristic on a given optimization problem provides balance between exploitation of the accumulated search experience and the exploration of the search space to identify regions with high quality solutions in a near-optimal way [4].

Metaheuristics algorithms are based on either trajectory search process or population based strategies. The trajectory metaheuristics explores the solution space by iteratively improving the current best solution in order to find the optimal one. The population based metaheuristics are nature inspired and use the population concept. They can be divided in two classes: evolutionary computing that utilize the models of species evolution and swarm based that adopt the social behaviour models. Examples of metaheuristics include simulated annealing, tabu search, iterated local search, evolutionary computation including genetic algorithms, ant colony optimization [5].

Swarm metaheuristics use computational and behavioural metaphor for solving combinatorial optimisation problems inspired by the collective behaviour of biological species and communities in nature as social insects (bees, termites, ants, wasps), vertebrates (herds, flocks, fish passages), etc. [6]. Natural collective and distributed intelligence can be observed at different biological levels as cells, organs, immune and nervous systems, living organisms. The general behavioural features of the swarm system can be regarded as a distributed community of autonomous individuals

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

(agents): the control is distributed between the agents; the communication between the individuals is local; decisions are taken by random agents; the system level behaviour is exceeds the individual agentbehaviour; the agents interact following simple rules.

The swarm systems are characterized by stability, adaptability, self-organization, scalability. Swarm based metaheuristics can be used for design of complex adaptive systems that comprise global level intelligence as a result of local interactions of the individuals.

The extremely fast evolution and growing availability of high performance computing platforms pose the requirement for efficient parallel implementations of the algorithms that solve complex problems utilizing distributed and shared memory multiprocessor parallel platforms. Parallelization of the metaheuristics has two main goals: diversification that gives the opportunity to explore concurrently various search space sub-areas and acceleration of the time consuming search process.

The paper presents parallel computational model for solving the time tabling problem using artificial bee colony (ABC) optimization. The suggested solution exploits a hybrid metaheuristics approach that combines the general ABC algorithm with some further trajectory based local search procedure following the simulated annealing concept. The suggested solution is targeted for multicomputer high performance architecture and exploits both shared memory and distributed memory parallel computational models utilizing fine grained thread level parallelism and master-slave message passing flat model. The experimental evaluation of the suggested solution is made base on C++ programming implementation using MPI library and OpenMP API. In addition a web based application is developed to ease the schedule construction, editing, visualization and usage in educational institutions.

II. TIME-TABLING PROBLEM

A. *Problem statement*

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 [7]. The constraints impose some restrictions and can be regarded as hard and soft.

Curriculum scheduling problem requires a weekly time table to be generated for planning lectures to be held in given number of available rooms during certain time slots. The problem requires the following considerations:

- *Time period and time slots*: Time periods are specified for given number of teaching days per week (usually 5 or 6). Each day is divided into a fixed number of time slots with an assumption that the time intervals are same for all days. Thus eachtime period is represented as a pair (day, time slot).
- *Lectures and lecturers*: Each curriculum consists of a number of lectures to be assigned to particular time periods. Each lecture is attended by a number of students and is taught by given lecturer. A minimum number of days per week in which each lecture will be given is specified in advance. In addition periods might be specified in which no classes should be assigned.
- *Rooms*: Each room has a capacity and all rooms are considered equally suitable for eachlectureprovided the required capacity is satisfied.
- *Curriculum*: The curriculum is composed of a set of lectures so that each pair of lectures in the set have common students.

The solution should also satisfy some predefined constraints. The constraints are regarded as hard and soft depending on the possibility none of them or some of them to be violated.

Among the hard constraints are the following:

- *Lectures*: All lectures should be assigned to different time periods. Violation of this constraint occurs if a lecture is not included in the schedule.
- *Room occupation*: Two lectures cannot be held in the same room in the same time period. Violation of this constraintoccurs if two classes are assigned to one and the same room at the same time. Each other lecture that is in conflict and is assigned to the same room and time period counts as another violation.
- *Conflict*: Two lectures in the same curriculum or given by the same lecturer should be assigned to different time periods. Conflict of two lectures assigned to one and same period is considered as constrain violation. Three conflicting lectures are counted as three violations, one for each pair.
- *Availability*: If a lecturer cannot give a lecture at certain time period, then no lecture of this lecturer should be scheduled for this period. Each lecture assigned to invalid period is considered as a violation.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

The soft constraints might include some of the following:

- *Room capacity*: The number of students who attend particular lecture must be less than or equal to the number of seats in the room to which the lecture is assigned. Each student above the specified number is considered as one violation penalty point.
- *Minimum number of training days*: The lectures must be assigned in a minimum number of days. Each day below the minimum is considered as 5 violation penalty points.
- *Curriculum compactness*: The lectures in one and the same curriculum should be assigned to consecutive time periods. Each lecture that is not following or preceding another lecture from the same curriculum is considered asa violation and adds 2 penalty points.
- *Room consistency*: Each lecture must be assigned to one and the same room. Each different room used for the lectures in given curriculum counts as one penalty point.

The solution of the curriculum schedule problem is given in the form of assignment of a period (day and time slot) and a room for all lectures in the curriculum.

B. *Related work*

Being a common scheduling problem the timetabling problem has been researched intensively providing various timetabling solutions that attempt to either suggest exact or near optimal solution. The optimization strategies based on metaheuristics are generally targeted at using populationbased approaches as genetic algorithms, ant colony optimization or hybrid memetic algorithms or trajectory based approaches as tabu search, simulated annealing,variable neighborhood search. Recentresearch directions in timetabling are described in [10, 11]. The Artificial Bee Colony (ABC) algorithm is one of the newest population based metaheuristics using a nature inspired metaphor for solving hard optimization problems [12]. Several researchers have used ABC for finding near optimal solution of numerous scheduling problems including job shop scheduling and timetabling problem. Most of the attempts are either based on application of ABC for examination timetabling [13-15] or improvement of search space capacities of the algorithm [16, 17].

III. ARTIFICIAL BEE COLONY OPTIMISATION FOR SOLVING THE TIME TABLING PROBLEM

A. *Artificial Bee Colony Algorithm*

The Artificial Bee Colony (ABC) algorithm is one of the newest metaheuristics. It is a swarm based metaheuristic algorithm for optimizing numerical problems that is inspired by the intelligent foraging behaviour of honey bees [8]. ABC is a population-based search procedure that can be used for solving optimization problems by combining both global exploration and local search of the solution space for finding near optimal solution to the problem in a reasonable time.

The population comprises several individuals (artificial bees) that are aimed to discover foods positions with high nectar amount in a multidimensional search space. The goal is to evaluate the food sources and at the end to provide the optimal solution to the problem solved as being the food source of highest nectar. The individuals are divided in three groups: employed and onlooker bees are searching for a food source in the space based on a global and local experience while the scouts choose the food sources randomly without using experience. ABC system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation process [9].

The general procedure of the ABC optimisation is as follows:

repeat
    each employee bee visits a food source stored in its local memory, finds adjacent food source, evaluates it and "dances" to the hive
    each onlooker bee observes the "dance", selects a food source according to the "dance", visits and evaluates it
    each scout bee selects some food sources to be abandoned and randomly replaces them with new sources
    the best food source is determined
until the requirements are met

Each food source position represents a possible solution of the optimization problem. The amount of nectar at given food source represents the quality of the proposed solution. The number of employee bees is equal to the number of the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

solutions in the population. At first initialpopulation is randomly generated. The main iterative procedure of the algorithm involves a search of the solution space by the employee, observer and scout bees.

The employee bees explores locally the search space adjacent to the position of the current solutionstored in their local memory and replaces it with the new best solution. To share its current bestsolution each employee bee completes a “dance” in order to attract onlooker bees to further explore its position. Each onlooker bee evaluates the fitness of the employee bee solutions and selects a solution for further exploration. As the employee bees, each onlooker compares the fitness of the new source with its current local and stores the new best in its local memory. If given food source is evaluated as nectar rich then more onlooker bees explores it. Scout bees identify food sources to be abandoned and randomly choose new ones to replace them. A food source is abandoned if the number of employee bees that find better solution in its position decreases and gets smaller than predefined parameter. The idea of leaving some food sources and replacing them by completely new position in the search space to be explored supports the local minima escape.

In the solution described in the paper a modification of the general ABC procedure is suggested that involve a trajectory based local modification of the explored solutions inspired by the simulated annealing metaheuristics. At each iteration the solution of the scouts are modified and given number of new solutions are generated, each one based on the previous one. The employee bees evaluate the new solutions and stores the current best in its local memory.

B. *ABC for the timetabling problem*

To apply the ABC algorithm for solving the timetabling problem the solution space is regarded as comprised by only the curricula schedules that fully satisfy the hard constraints. Each solution is evaluated by the bees in the population based on the violation of the soft constraints. In order to generate adjacent solution a modification in given timetable is made by exchange of the scheduled periods of two lectures.

The algorithm comprises the following steps:

```
Scout()
while unexplored positions in the search space
    s ←lookForNectarHere()
    evaluatePosition(s)
    if good enough
        danceToCallWorkers()
    while food
        Worker()
            s2, s3, s4 … ←ChangePositionSlightly()
            evaluatePosition()
            s ←determineBestSolution(s2, s3, s4…)
            AbandonFoodSource()
        end Worker
    end while
end while
while termination criteria not satisfied
    Worker()
        s2, s3, s4 … ←ChangePositionSlightly()
        s ←determineBestSolution(s2,3,4…)
        end Worker
    end while
```

A modification of the general ABC procedure is suggested that involve a trajectory based local search of the explored solution space inspired by the simulated annealing metaheuristics. At each iteration the solutions of the scouts are modified and given number of new solutions are generated, each one based on the previous thus making local moves in the search space. A solution modification is based on random selection of two lectures and exchange of their scheduled periods. According to the simulated annealing concept the generated moves that reduce the cost function are accepted with high probability whereas those that increase the cost are accepted with low probability. The employee bees evaluate the new solutions generated by the scouts, explores their local neighbourhood and stores the current best

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

solution in the local memory. The suggested hybrid approach involving local moves in the search space allows better diversification of the ABC approach and prevents the algorithm from falling in a local minimum in the solution space.

IV. PARALLEL COMPUTATIONAL MODEL FOR SOLVING THE TIME TABLING PROBLEM USING ABC OPTIMISATION

In order to reduce the computational time a parallel model is suggested that is targeted to a high-performance computer platform that allows parallelism to be exploited both at a shared memory and distributed memory levels. The thread level parallel processing for shared memory platform uses a data parallel computations utilized as thread level fork-join construct for simultaneous work of several scout bees (fig 1). The model requires each scout to provide solution exploration for its own employee bees without data sharing between the scouts. Sharing a common resource, the employee bees in this case, would unnecessarily complicate the parallel algorithm without any substantial benefit from it.

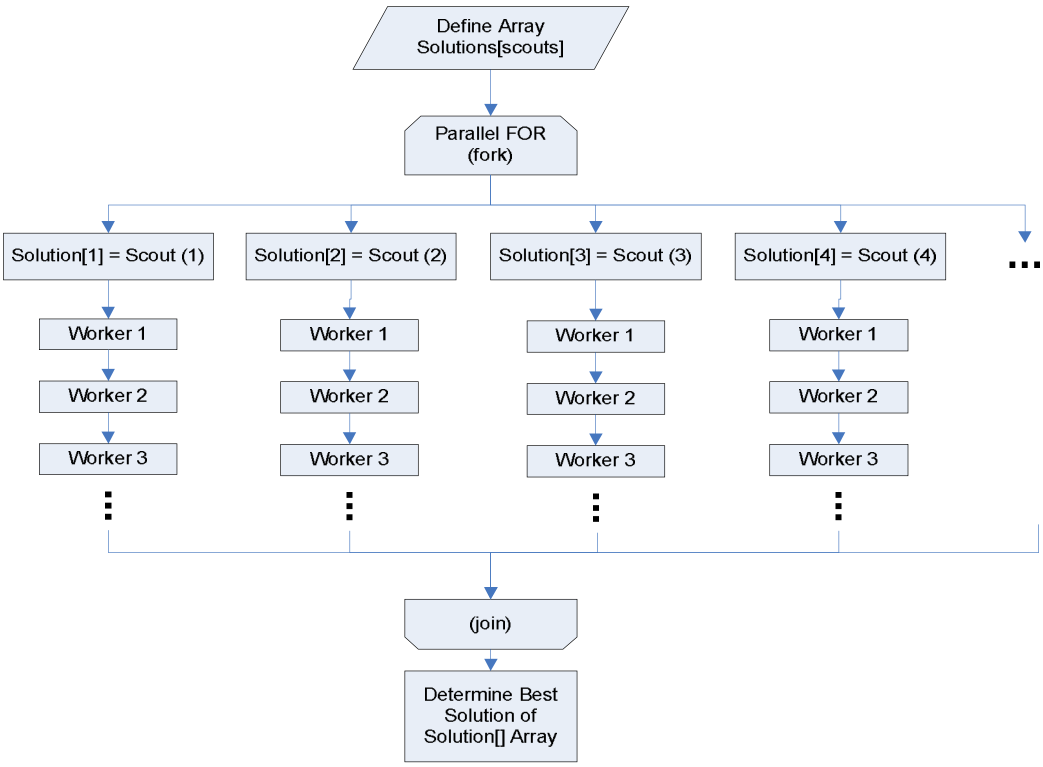

Fig.1. Thread level parallel computational model of scout bees

Additional acceleration and diversification of the metaheuristics algorithm is achieved by implementing a master-slaveparallel model in a multicomputer cluster platform. Each slave processexecutes the thread parallel model for scout computations and then sends the best solution to the master process (fig.2). The results for the generated best curricula schedule is given as a timetable of the courses assigned to each room for each time period (Table 1).

The performance analysis shows that the parallel computation model scales well on the multicomputer platform – the speedup increases almost linearly with the machine size (fig. 3). The system is well balanced and the computational workload is evenly distributed throughout the system considering the client processes.

The experimental study for the impact of the suggested parallel computational model on the solution quality of the ABC optimisation is based on 50 runs on the multicomputer platform (fig. 4). The results show that the overall average solution quality scales well with the increase of the machine size due to the better diversification provided by the increased number of the processes as well as the trajectory based local search.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

The speedup of the parallel computational model is shown on fig 5. The results show good scalability of the parallel computational model in respect to both the parallel workload (number of iterations of the optimisation algorithm) and the size of the cluster.

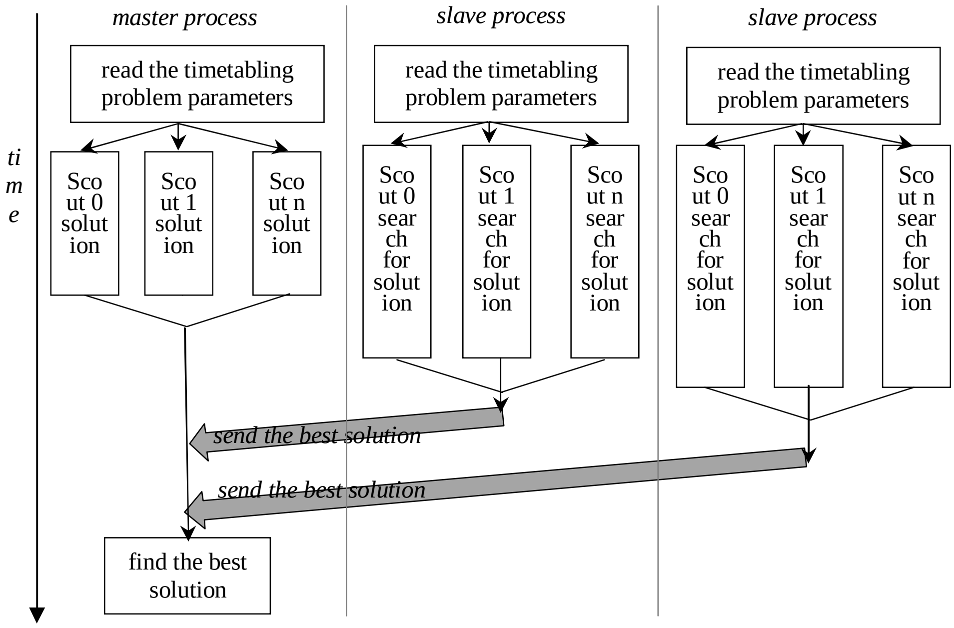

Fig.2. Sequence diagram of the master-slave parallel computational model

Table 1. Best solution generated by the ABC optimisation

Room ID: r25	*1*	*2*	*3*	*4*	*5*
Mon	*c0965*			*c0821*	*c0857*
Tue	*c0433*	*c0533*	*c0965*	*c0433*	*c0965*
Wed		*c0857*			*c0018*
Thu		*c1067*	*c0018*	*c0855*	*c0965*
Fri		*c0855*	*c0857*	*c0965*	*c0533*

Room ID: r26	*1*	*2*	*3*	*4*	*5*
Mon	*c0951*	*c0826*	*c0855*	*c0793*	
Tue	*c0027*	*c0816*	*c0442*	*c1020*	
Wed		*c1068*	*c0826*		*c0951*
Thu		*c0027*	*c0855*	*c0442*	
Fri		*c1020*	*c0816*	*c0855*	

…

Overall solution penalty: 975

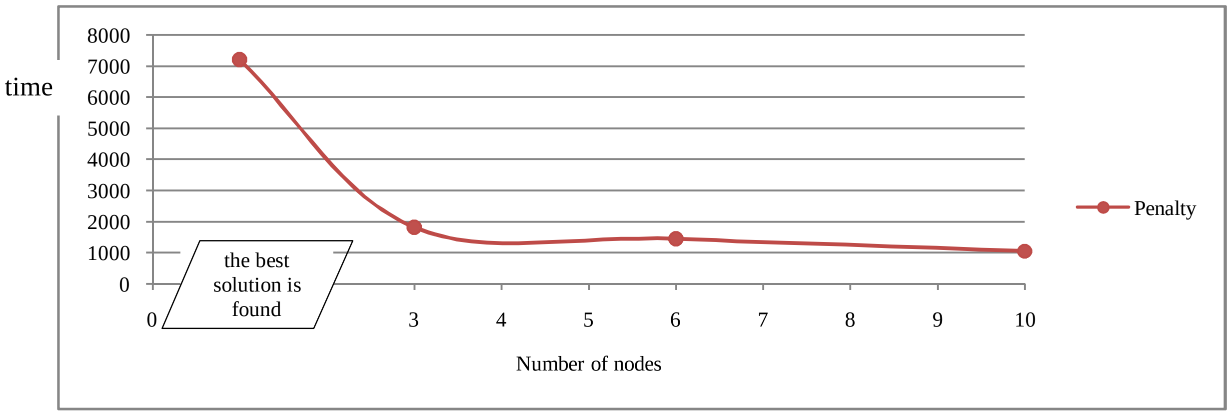

Fig.3. The overall average solution penalty based on 50 independent runs

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

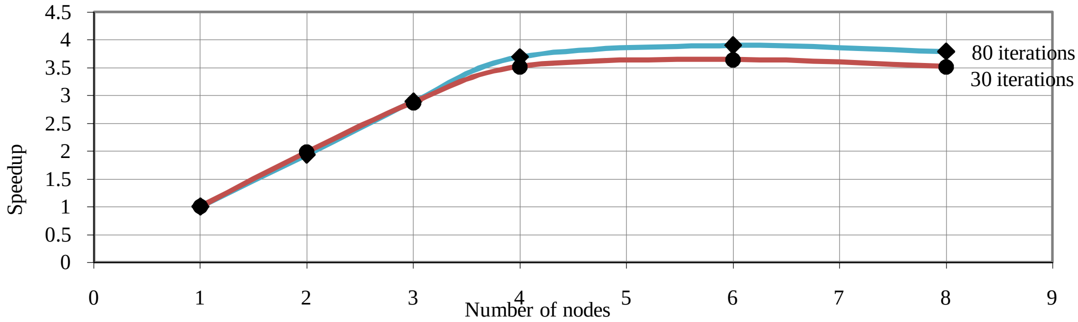

Fig.4. Speedup of the parallel computational model

V. Web Based Application for Solving the Time-Tabling Problem

A web based application is also developed in order to utilize the described approach for solving the timetabling problem using ABC optimization. The application provides the following possibilities:

- user interface for easy definition of the curricula parameters;
- automatic generation of the best solution using the ABC optimization
- visualization of the generated schedule;
- manual manipulation of the generated schedule;
- archive of the generated solution;
- using several instances for different curricula.

The basic architectural model of the application is presented on fig. 6.The application consists of four main modules: user interface, web server, database management system and an ABC schedule generation module.

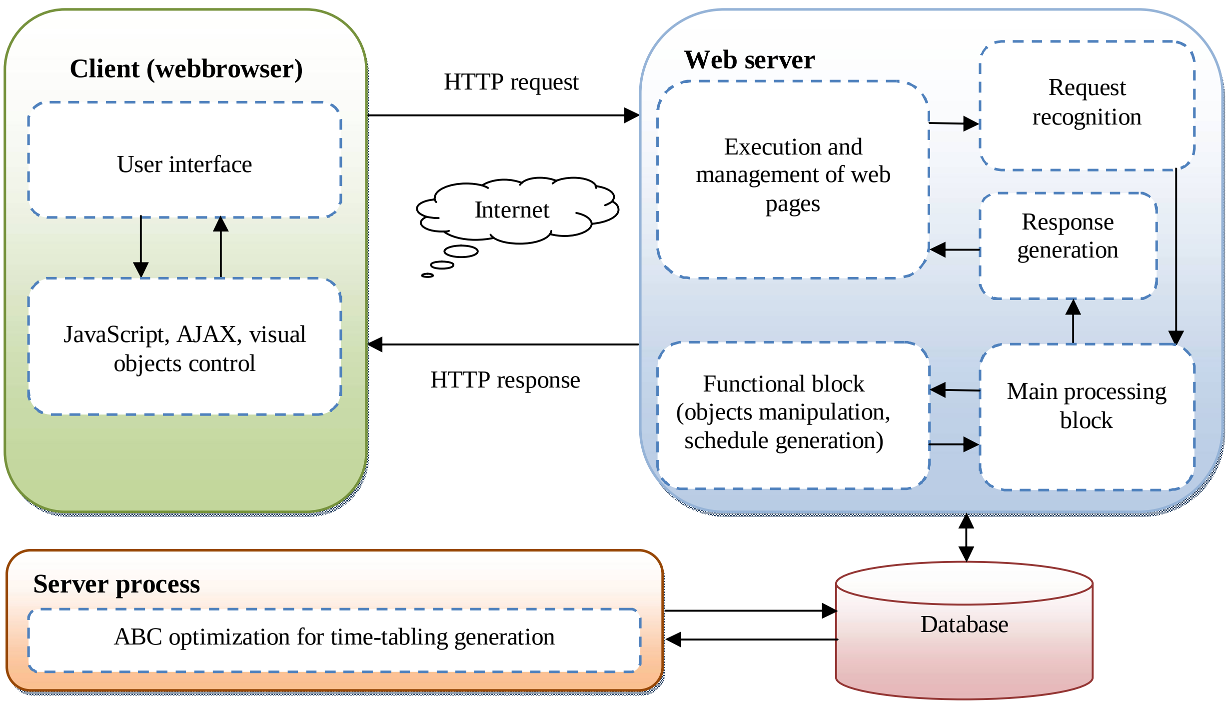

Fig. 6. Architectural model of the application

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

VI. CONCLUSION AND FUTURE WORK

The paper presents parallel computational model for solving the curricula time tabling problem using hybrid metaheuristics approach that combines population based artificial bee colony optimization and trajectory based simulated annealing for local optimization. The suggested solution is targeted for multicomputer high performance architecture and exploits both shared memory and distributed memory parallel computational models utilizing fine grained thread level parallelism and master-slave message passing flat model. The experimental evaluation shows good scalability of the solution quality due to better diversification and local exploration of the search space when increasing the number of processes and threads. The speedup of the parallel computational model also scales almost linearly in respect to both the parallel workload and the machine size. In addition a web based application is developed to ease the schedule construction, editing, visualization and usage in educational institutions.

REFERENCES

1. Klein, P. N. , Young, N. E, "Approximation Algorithms for NP-hard Optimization Problems", Algorithms and Theory of Computation Handbook, CRC Press, 1999.
2. Dréo, J., Pétrowski, A., Siarry, P., Taillard, E., "Metaheuristics for Hard Optimization", Springer-Verlag, Berlin, 2006.
3. Blum, C., Roli, A., "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison", ACM Computing Surveys, Vol.35, №3, pp.268-308, 2003.
4. Birattari, M.,Paquete, L., Stützle, T., Varrentrapp, K., "Classification of Metaheuristics and Design of Experiments for the Analysis of Components", Technical Report AIDA-01-05, FachgebietIntellektik, FachbereichInformatik, TechnischeUniversität Darmstadt, 2001.
5. BoussaïD, I., Lepagnot, J.,Siarry, P., "A Survey on Optimization Metaheuristics", Information Sciences, Vol.237, pp.82-117, 2013.
6. ClercM., "Particle Swarm Optimization", ISTE, UK, 2006.
7. Badri, M. A., Davis, D. L., Davis, D. F., Hollingsworth, J., "A Multi-Objective Course Scheduling Model: Combining Faculty Preferences for Courses and Times", Computers and Operations Research, Vol.25, pp.303-316, 1998.
8. Karaboga, D., Basturk, B., "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", Journal of Global Optimization, Vol.39, №3, pp.459-171, 2007.
9. Karaboga, D., "Scholarpedia", Vol.5, №3:6915, 2010.
10. Pillay, N., "A Survey of School Timetabling Research", Annals of Operations Research, Vol. 218, №1, pp.261-293, 2014.
11. McCollum, B., "A Perspective on Bridging the Gap Between Theory and Practice in University Timetabling", Practice and Theory of Automated Timetabling VI, Lecture Notes in Computer Science, Vol.3867,pp.3-23, 2006.
12. Lewis, R., "A Survey of Metaheuristic-Based Techniques for University Timetabling Problems", OR Spectrum, Vol.30, №1, pp.167-190, 2008.
13. Alzaqebah, M., Abdullah, S., "Artificial Bee Colony Search Algorithm for Examination Time Tabling Problems", International Journal of the Physical Sciences, Vol.6, №17, pp.4264-4272, 2011.
14. Alzaqebah, M., Abdullah, S., "Hybrid Artificial Bee Colony Search Algorithm Based on Disruptive Selection for Examination Timetabling", Proc. of 5th International Conference COCOA 2011, Zhangjiajie, China, pp.31-45, 2011.
15. Bolaji, A. L., Khader, A. T., Al-Betar, M. A., Awadallah, M. A., Thomas, J. J., "The Effect of Neighborhood Structures on Examination Timetabling With Artificial Bee Colony", Practice and Theory of Automated Timetabling (PATAT 2012), Son, Norway, pp.131-144, 2012.
16. Sabara, N. R., Ayoba, M., Kendallb, G., Qub,R., "A honey-Bee Mating Optimization Algorithm for Educational Timetabling problems", European Journal of Operational Research, Vol.216, №3, pp.533–543, 2012.
17. Bolaji, A. L., Khader, A. T., Al-betar, M. A., Awadallah, M. A., "An Improved Artificial Bee Colony for Course Timetabling", Proc. of Sixth International Conference Bio-Inspired Computing: Theories and Applications (BIC-TA), pp.9-14, 2011.

BIOGRAPHY

Milena Lazarova is an Associate Professor at the Department Computer Systems, Faculty of Computer Systems and Control at the Technical University of Sofia. She received a PhD in 1999 anda MSc. degree in 1995 from the Technical University of Sofia, Faculty Computer Systems and Control. Her research interests are in the field of parallel information processing, parallel algorithms, parallel programming, metaheuristics, image processing and analyses, computer graphics, pattern recognition.