The ParMetaOpt Experience: Performance of Parallel Metaheuristics on Scheduling Optimization

PLAMENKA BOROVSKA, GEORGE YANCHEV
Computer Systems Department
Technical University of Sofia
8 Kliment Ohridski Boul., 1756 Sofia
BULGARIA
pborovska@tu-sofia.bg http://csconf.org/cs/leader_eng.htm
jyanchev@dir.bg

Abstract: Parallel metaheuristics provides innovative and powerful alternative for combinatorial optimization providing the opportunity to find out near-optimal solutions in reasonable time. The goal of this paper is to reveal the experience of utilizing the experimental parallel metaheuristics framework ParMetaOpt, developed at Computer Systems Dept., Technical University of Sofia. Parallel metaheuristics algorithm have been designed and implemented based on population based methods (evolutionary computation, artificial bee colony and ant colony optimization) and trajectory based methods (GRASP, Tabu search and simulated annealing) for the case studies of the timetabling and the job shop scheduling problems. Parallel performance evaluation and analysis have been presented on the basis of hybrid (MPI+OpenMP) parallel program implementations on compact heterogeneous computer cluster.

Key-Words: Metaheuristics, Combinatorial Optimization, Scheduling Optimization, Parallel Algorithms and Programming, Parallel Performance

1 Introduction

Modern metaheuristics [1,4] has emerged as an innovative and powerful alternative that offers a wide spectrum of algorithmic frameworks providing the opportunity for IT professionals to solve combinatorial (NP-complete) problems in reasonable time and finding out solutions of good quality. Parallelizing metaheuristics [2] is a grand challenge due to the fact that the fundamental concepts of metaheuristics – diversification and intensification – predetermine the possibility to deploy parallel computing strategies to obtain better performance and to improve the quality of solution as well.

ParMetaOpt is an experimental parallel metaheuristics framework [3], developed at Computer Systems Dept., Technical University of Sofia, that is intended to provide flexible components for implementing parallel metaheuristics algorithms for combinatorial optimization based on object oriented programming techniques. The computing platform comprises heterogeneous compact computer cluster comprising dual-core Opteron-based servers and quad-core Xeon-based servers, the operating system is Scientific Linux, the programming language is C++ in combination with the standard application programming interfaces MPI and OpenMP. The experimental framework is used by researchers, PhD students and MSc students.

The goal of this paper is to share our experience in solving scheduling optimization problems such as the timetabling and the job shop scheduling problems on the ParMetaOpt parallel metaheuristics framework in respect to parallel performance.

2 Scheduling Optimization Problem Instances

Scheduling optimization problems are hard and time consuming to solve. Scheduling problems deal with the temporary planning of activities with limited resources. Depending on the specific problem the activities may be jobs, tasks, lectures, trains, etc. The corresponding resources may comprise machines, processors, teachers, etc. The case studies of scheduling optimization under investigation are the course timetabling problem and the job shop scheduling problems

2.1 The course timetabling problem

The course timetabling problems require the assignment of a set of events such as lectures, labs, etc. to a limited number of time slots satisfying predefined constraints. The solution suggests a sequence of meetings between teachers and students in a prefixed period of time (typically a week) in the available classrooms and auditoria, satisfying a set of constraints of various types
(dependencies between the activities) such as fixed period of day time, the capacity of classrooms, the maximum limit of time per day for the students and lecturers, the equipment of the classrooms, etc.

A predefined number of classes \( E \) have to be scheduled in timeslots of 45 min (5 days, 9 hours per day), to a set of auditoria \( A \), a set of students \( S \) who attend lectures and labs, and a set of equipments satisfied by the auditoria and required by lectures, labs, etc. The hard constraints include that no student can attend more than one lecture or lab at a time, the auditoria or lab has enough capacity and the equipment required, in each auditorium only one event is allowed for a specified timeslot, etc. A solution is feasible if no hard constraint has been violated. The goal is to minimize the number of violated soft constraints in the time preferences of resources and activities.

The timetabling problem has been vastly investigated utilizing metaheuristics methods [5]. The experimental results have shown similar range of the solutions quality obtained [6]. The goal of the investigation is to evaluate the speedup of solving the course timetabling problem with parallel metaheuristics.

### 2.1.1 The job shop scheduling problem

Job shop scheduling (JSS) is a combinatorial optimization problem in which jobs are assigned to resources in particular times. We are given a set of jobs \( J_1, J_2, ... , J_n \) of varying sizes, which need to be scheduled on a set of machines, while trying to minimize the total length of the time. The problem variations may be summarized as follows: machines can be related, independent, or identical, a certain time gap may be required between jobs, machines can have sequence dependent setups, jobs may have constraints (sequence dependencies, for ex.), jobs and machines may have mutual constraints (certain jobs can be scheduled only on a predefined subset of machines).

The JSS problem has been vastly investigated utilizing metaheuristics methods [4]. The experimental results have shown similar range of the solutions quality obtained. The goal of the investigation is to evaluate the speedup of solving the JSS problem with parallel metaheuristics.

### 3 Methodology for Parallel algorithm Design

The methodology for the design of parallel algorithms and the relevant parallel computational models and their parallel program implementations is based on the correlation analysis of the architectural and algorithmic design spaces. The major advantage of this methodology is that the architectural specifics of the target computing platform are taken into account in the early stages of the design process. The compact heterogeneous cluster comprises 8 dual-core Opteron based servers and 2 quad-core dual-processor Xeon based servers. Actually, from architectural point of view the computing platform represents a cluster of symmetric multiprocessors (a cluster of SMP’s).

The focus of the methodology for parallel algorithmic synthesis is the correlation analysis of the algorithmic space (the abstract machine model – semantic and performance attributes) and the architectural space (the architectural model of the target computing platform). Fundamental issue is to adapt the inherent granularity of the application to the optimal granularity of the platform determined by the architectural model. The output of the abstract synthesis is the task graph of the parallel application. The focus of the practical synthesis is on resource scheduling i.e. the task graph of the parallel application is mapped on the graph model of the parallel machine. As a result of the abstract and practical synthesis the parallel computational model is constructed. Finally, the parallel algorithm performance and scalability are evaluated on the basis of measuring and profiling the relevant parallel program implementation.

![Fig. 1. The methodology for parallel algorithm design.](image)

In the case of parallelizing metaheuristics algorithms for a specified problem (instance) the following procedure is performed:
1. Metaphor specifics analysis
2. Simulation model design for the problem to be solved (instance)
3. Parallelization (define strategy, parallel paradigm and parameter tuning)
4. Improvements to accelerate the convergence of the parallel algorithm
5. Parallel program implementation development
6. Parallel performance evaluation and analysis
7. Quality of Solution evaluation and analysis

Experiments have been made for three population based metaheuristics – evolutionary computations (EC), Swarm (Artificial Bee Colony - ABC), and Ant Colony Optimization (ACO) as well as for three trajectory based metaheuristics – Tabu Search, Greedy Randomized Adaptive Search Procedure (GRASP) and Simulated Annealing (SA).

For the case of evolutionary computations the parallel implementation is based on the island model, while the parallel paradigm is Single program multiple data (SPMD). The improvements applied to accelerate the convergence of the parallel algorithm are elite chromosome migration and local evolutions with parallel mutation rates.

Swarm intelligence is presented by the Artificial bee colony optimization. The parallel paradigm selected is manager/workers. The parallel computations follow the bulletin-board model. If the quality of the current solution degrades the number of the worker bees is reduced and the source is abandoned.

The parallelizing strategy for ACO is to deploy Ant Colony Systems (ACS). The parallel paradigm is SPMD.

The improvements incur elite ants’ migration and virtual ring neighborhood.

For the case of GRASP the parallel implementation is based on the parallelization strategy “multiple walks – independent threads”, while the parallel paradigm is manager/workers. The manager is responsible to select the best solution. The improvements deployed to accelerate the convergence of the parallel algorithm are path – relinking and don’t look bit technique.

Parallel Tabu search implies control cardinality of the type “pC” i.e. multiple cooperating processes. Each process performs individual search and communicates with other processes. Parallel search is terminated when all process have finalized. The parallel paradigm applied is of the SPMD type. The improvements include search intensification and temporal memory utilization.

The parallelizing strategy for simulated annealing implies multiple independent runs (MIR) in respect to coarse granularity level and parallel moves.
(parallelization of Markov chains) in respect to fine granularity level. For all types of parallel metaheuristics under investigation we have developed hybrid parallel program implementations (MPI + OpenMP based).

For all types of parallel metaheuristics under investigation we have developed hybrid parallel program implementations (MPI + OpenMP based).

Fig. 3. Parallelizing trajectory based metaheuristics for scheduling optimization

4 Experimental Results and Analysis
A number of experiments have been carried out on the experimental metaheuristics framework ParMetaOpt considering the two scheduling optimization problems.

The speedup is defined and evaluated as the ratio of the time for executing the parallel code on a single processor to the time for its parallel execution.

The speedup obtained for the case of the course timetabling problem is shown in Fig.4, while for the case of the job shop scheduling problem the results are shown in Fig.5.

The analysis of the parallel performance for both optimization problem instances under investigation show that the population-based metaheuristics (ACS, ABC and Evolutionary computing) provide satisfactory speedup. For the case of simulated annealing, however, a slowdown of the system performance is observed i.e. the time for parallel computing is greater than the time for sequential processing. This anomaly is explained by the fact that the parallel paradigm is multiple independent runs (MIR) and the stopping criteria imply specified convergence level of the algorithm.

Fig. 4. Speedup of parallel metaheuristics for the timetabling problem on the compact heterogeneous computer cluster.

The time for parallel computing is determined by the process of SA that converges last. The parallel optimization finishes when the slowest process converges and terminates. In the case of parallel...
simulated annealing the advantages of the parallel search imply the higher level of the search diversification i.e. parallel search in various areas of the search space.

![Parallel Performance Graph](image)

**PARALLEL PERFORMANCE**

Fig.5. Speedup of parallel metaheuristics for the job shop scheduling problem on the compact heterogeneous computer cluster.

### 4 Conclusion and Future Work

In this paper we present the experience of utilizing the experimental parallel metaheuristics framework ParMetaOpt, developed at Computer Systems Dept., Technical University of Sofia. Parallel metaheuristics algorithms have been designed and implemented based on population based methods (evolutionary computation, artificial bee colony and ant colony optimization) and trajectory based methods (GRASP, Tabu search and simulated annealing) for the case studies of the timetabling and the job shop scheduling problems. Parallel performance evaluation and analysis have been presented on the basis of hybrid (MPI+OpenMP) parallel program implementations on compact heterogeneous computer cluster comprising dual-core and quad-core servers.

Parallel performance analysis shows that the parallel metaheuristics under investigation – population based metaheuristics such as ACS, ABC and evolutionary computations as well as trajectory based methods such as GRASP and Tabu search yield satisfactory parallel performance for both case studies (timetabling and job shop scheduling). In the case of parallelizing simulated annealing a slowdown speedup anomaly is observed due to the sequential nature of the computational model and the difficulties in parallelizing serial Markov chains. In case of the parallel simulated annealing the advantages are expected to be due to the increased level of diversification.

The future work implies developing agent-based technologies for implementing parallel metaheuristics targeted at solving scheduling problems.

### ACKNOWLEDGEMENTS

This work represents the results of the research project “Experimental High Performance Platform for Combinatorial Optimization of Science and Engineering Problems Based on Parallel Metaheuristics”, Grant 091хп153-09, funded by the Research sector, Technical University of Sofia.

### References:


