Efficiency and Quality of Solution of Parallel Simulated Annealing

PLAMENKA BOROVSKA, MILENA LAZAROVA
Department of Computer Systems
Technical University of Sofia
8 Kliment Ochridsky str.
BULGARIA
http://www.tu-sofia.bg/Bul/faculties/fksu/ks/index.htm

Abstract: - The paper describes parallel computations for solving room assignment problem using simulated annealing on multicomputer platform. Parallel independent runs on all available processors are suggested providing both speedup of the calculations of the best final arrangement and high diversification of the independent moves in the state space. On the base of experiments with parallel MPI implementation both speedup and quality of solution of the suggested parallel simulated annealing algorithm are examined and analyzed. The impact of multicomputer size (number of parallel working workstations) and workload (number of persons) on the parallel performance are explored and performance profiling and analysis are presented.

Key-Words: - Parallel Metaheuristics, Simulated Annealing, Room Assignment Problem, Parallel Computing, Multicomputer, Message Passing

1 Introduction

Simulated annealing (SA) is a Monte Carlo based method for numerical optimization that implies the principles of thermodynamics and is motivated by an analogy to annealing in solids [1]. It performs optimization without prior knowledge of the problem structure or of any particular solution strategy. SA has been applied for solving numerous practical and theoretical tasks including many NP problems and applications in the areas of signal and image processing, task allocation, course scheduling, network design, graph coloring and partitioning, molecular analysis [2, 3, 4, 5, 6].

The room assignment problem is a combinatorial optimization problem that is very similar to the timetable scheduling problem. The goal is to accommodate given number of persons in twice less number of rooms minimizing the conflicts between the room mates. Similar optimization techniques include metaheuristics such as tabu search, genetic algorithms, GRASP [7, 8, 9].

There are two motives to use parallel metaheuristics. One motive is diversification that gives the opportunity to explore concurrently various search space sub-areas. The other motive is intensification that implies intensive search in regions promising high quality solutions.

The paper is aimed at investigating the efficiency and quality of solutions of parallel simulated annealing on multicomputer platform for the case study of the room assignment problem. The experimental study is based on parallel independent runs implementing various parallel random generators, initial temperatures and stopping criteria. In our experimental study we have used linear congruential random number generators and parallel random number generators constructed on the basis of the leapfrog method. The impact of various factors such as the stopping criteria (number of iterations, the length of the chain of consequent unaccepted changes) as well as the initial temperature on the quality of solutions obtained have been investigated and estimated. The speedup and quality of solution of parallel simulated annealing have been estimated on the basis of flat parallel program implementations on a multicomputer platform. Parallel performance profiling and analysis are presented. Scalabilty of the parallel systems has been estimated in respect to the size of the multicomputer (number of workstations) and application size (number of persons).

2 Simulated annealing and the room assignment problem

The analogy between a physical many-particle system and a combinatorial optimization problem is based on the following equivalences [2]:

- solutions in a combinatorial optimization problem are equivalent to states of a physical system;
- the cost of a solution is equivalent to the energy of a state;
- a control parameter plays the role of temperature, such that:

- at high temperatures changes in energy are accepted;
- at low temperatures only decreases or smaller increases in energy are accepted;
- if temperature approaches zero no increases are accepted at all.

Furthermore a characteristic feature of SA is that there is no limitation on the acceptable size of an energy increase.

SA is used to solve complex optimization problems attempting to minimize some cost function. The algorithm is iterative. At each iteration a candidate solution to the problem under investigation is generated and its cost is calculated. The solution is then perturbed. If the perturbation results in a decrease (or no change) in the cost then it is accepted, if the new cost is greater, then the perturbation is accepted only with a certain probability $e^{-\Delta/T}$, where Δ is the difference between the old and new value of the cost function and T is referred as a temperature. At the beginning of the algorithm, there is a high probability that all increases in cost function are accepted providing relatively large moves in the solution space. The probability of accepting a solution leading to a cost increase is then decreased slowly throughout the iterations with the expectation that the algorithm will tend to stop at or near the minimum cost.

A temperature parameter is used to control the probability of acceptance of uphill moves – these are solutions that increase the cost function. Initially the temperature is set to a high value and most uphill moves are accepted. As the algorithm progresses the temperature is lowered and the probability of accepting uphill moves decreases. The number of moves at each temperature is referred to as "chain length" and can be regarded as Markov chain. Cooling schedule of the simulated annealing is determined by the chain length and the rate at which the temperature decreases from chain to chain:

$$T_N = f(T_0, T_N, N, n) \qquad (1)$$

where T_0 is initial temperature, T_N is end temperature and T_n is the temperature value at iteration n.

Given an appropriate temperature schedule and an infinite amount of time the algorithm will converge to the true minimum [10].

The cooling schedule is very important to the successful application of simulated annealing because if the temperature is decreased too rapidly, either by steps that are too large or chains that are too short, the algorithm may become trapped in a sub-optimal minimum and thus level off above the true minimum. If the temperature is decreased too slowly progress may be too slow and thus time-consuming. It may require a substantial amount of effort to determine an effective single-run schedule for any particular problem and this problem may be more complex than the original problem [10].

Practical implementation of the simulated annealing requires the following issues to be determined:

- the way of representing the solutions;
- the procedure for generating new solutions from the current solution;
- the cooling schedule;
- the cost function.

Room assignment problem attempts to arrange N persons in $N/2$ rooms minimizing the cost function defined as a sum of conflicts between the persons placed in one and the same room. The input data is a conflict table C of size $N \times N$ with value $C[i, j]$ interpreted as a dislike coefficient if person i and person j are placed together in one room. The table is symmetric with respect to the main diagonal.

Different methods and approaches have been studied for solving the above problem including application of logic programming and metaheuristics.

SA proved to be an effective method of solving optimization problems.

For the room assignment problem the SA algorithm is applied considering the following correlations [11, 12]:

- the solution to the problem is represented as assignment of each person to a given room;
- persons are placed in different rooms randomly;
- the cost function is simple sum of the dislike coefficients of all persons in the same room.

SA operates by swap of two randomly selected persons, i.e. exchanging their rooms. The change in cost that is caused as a result of such swap is computed. According to the SA algorithm a decrease in cost is accepted and an increase is only accepted if it satisfies a predefined probability equation. In order to compute the change in cost it is not necessary to reevaluate the cost of the entire current assignment, but only the change caused by the persons that change their rooms.

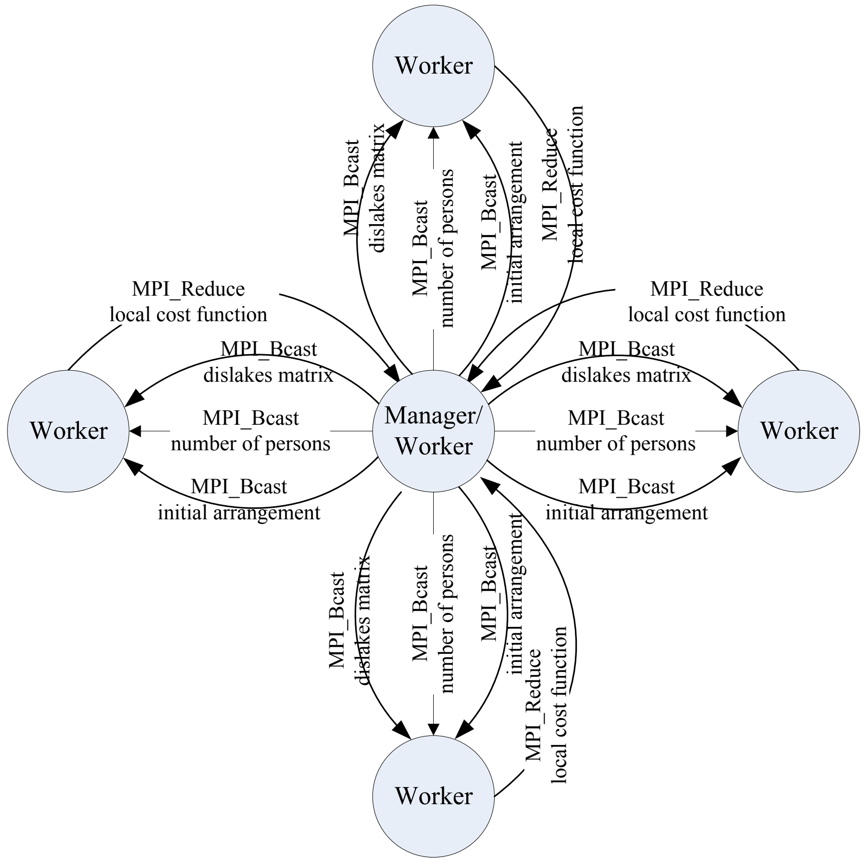

Fig.1. Parallel computational model of simulated annealing for the room assignment problem

3 Parallel Computational Model

Although simulated annealing is an efficient algorithm to solve problems it takes a long time to yield a good solution in some real case studies. The parallelization of the simulated annealing provides both speedup of the calculations attempting to find the best final solution and diversification of the moves in the search space.

The parallelization of simulated annealing for the room assignment problem is not straightforward. The problem itself does not permit a divide-and-conquer approach. The computationally parallelizable part of the problem, which is the cost function calculation, takes a relatively small portion of the runtime since it can be carried out incrementally.

The suggested parallel computational model for simulated annealing algorithm applied to room assignment problem is based on utilization of multiple independent runs by parallel working processes in multicomputer environment. Master-worker parallel programming paradigm is used and processes communication and synchronization is carried out by message passing among the processors (Fig.1).

The activities of the master process are the following:

- reads the input data (person dislikes matrix);
- generates different initial assignments equal to the number of parallel processes; the assignments are acquired by random swaps of persons' rooms;
- distributes the generated different initial assignments to all processes by global message passing communication, keeping one for itself (function MPI_Bcast);
- performs necessary computations working as a slave following the general algorithm of simulated annealing, described in the previous section, and makes number of moves in the solution space;
- gathers the results of the locally estimated cost functions by all worker processes (function MPI_Reduce) and determines the best value, i.e. the minimum sum of dislikes.

Each worker process is responsible for the following tasks:

- receives initial assignment from the master process;

- randomly selects persons to change their rooms and calculates the cost function after the swap;
- determines if the swap will be accepted or will be discarded;
- changes the temperature parameter according to predefined cooling schedule;
- sends the result of the current cost function to the master process when algorithm termination conditions are met.

We have experimentally evaluated the influence of different termination conditions on the performance parameters:

- fixed number of movements in the solution space;
- fixed number of consequent unaccepted changes (chain length), i.e. moves that increase the cost function;
- termination at a predefined value of the temperature.

In order to utilize the possibility for high diversification of the parallel computations parallel random number generators have to be utilized with the following features:

- the different generated sequences of random numbers should not be correlated;
- the generators have to be scalable providing independent sequences of random number for large number of processors;
- the parallel random number generators have to be local in terms each process can generate new sequence of random numbers without necessity of communications with other processes.

In applying parallel simulated annealing for the room assignment problem random number generator (RNG) is applied both at the stage of random selection of the persons to change their rooms and when evaluating the probability of acceptance of room change that increases the cost function.

We have explored the influence of the solution quality utilizing two different RNG – parallel RNG using leap-frog method and parallel linear congruential RNG with the number of process used as a seed value.

4 Analysis of Parallel Performance and Quality of Solution

The experiments of solving room assignment problem by parallel simulated annealing on multicomputer platform are performed on a multicomputer platform comprising 10 workstations (Intel Pentium IV 3.2GHz, 1 MB RAM, hyperthreading) connected by Fast Ethernet 100 Mbps switch. For the implementation of the parallel simulated annealing Microsoft Visual Studio 2005 and MPICH-2 are used.

The speedup gained by parallel independent runs according to the suggested computational model is presented in Fig.2. The results show almost linear speedup when increasing the number of parallel working processes. The speedup is larger for greater number of persons that is directly connected with greater computational workload in finding the final solution.

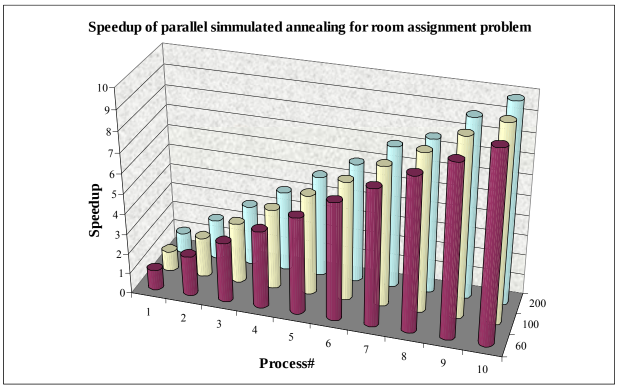

Fig.2. Speedup of parallel simulated annealing for different number of persons and different size of the multicomputer (scalability)

The results for the cost function obtained by parallel independent runs with different size of the multicomputer platform for 100 persons, leap-frog PRNG, and fixed number of 2500 moves are shown in Fig.3. Obviously, utilization of more parallel working processes provide better final solution for fixed number of iterations of the algorithm due to higher diversification of the state space searches provided by the parallel computations when parallel random generator guarantees uncorrelated sequences on the different processors.

The same tendency can be observed in Fig.4 where the dynamics of the values of the cost function for each of ten parallel working processes for 60 persons, 1360 iterations and leap-frog RNG are shown.

The impact of the initial temperature value on the time of the algorithm convergence is shown in Fig.5 and Fig.6. The influence of the multicomputer size, the PRNG used and the number of iterations for room assignment of 60 persons is shown in Fig.7. The influence of the termination criteria and the size of multicomputer for room assignment of 60 persons are shown in Fig.8.

The analysis of the results obtained show that the use of leapfrog parallel random number generators leads to better quality of solution because it provides low correlation of the generated sequences of random numbers. Parallel simulated annealing gives parallel exploration of sets of search paths in the solution state space and thus supplies high level of diversification.

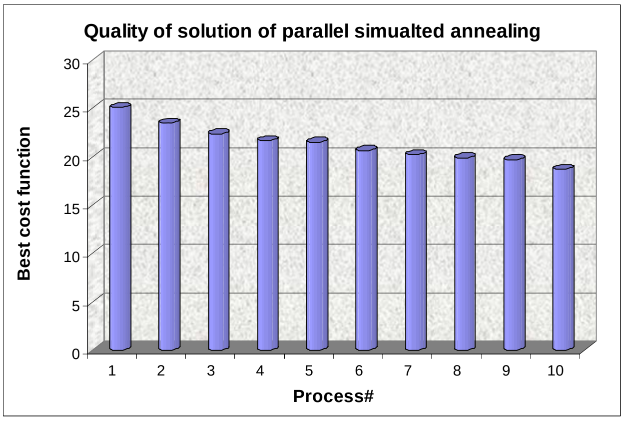

Fig.3. Quality of solution of parallel simulated annealing for room assignment of 100 persons

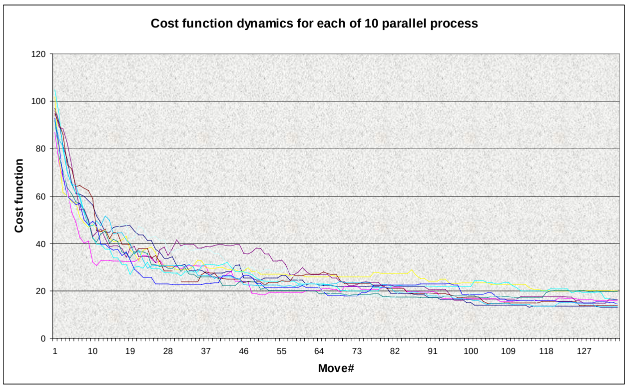

Fig.4. Cost function dynamics at each of 10 parallel running processes for 1300 moves

The solution quality is better for bigger parallel machine size and bigger parallel computational workload (the number of iterations), that means parallel simulated annealing scale well with respect to the size of the parallel machine and the size of parallel application. Apparently, the best approach is to use termination criteria that are determined by the convergence of the parallel algorithm. The benefits of parallel simulated annealing are observed even for the case of utilizing congruential RNG for 1500 iterations – the value of the cost function from 34.44 (sequential processing) dropped to 10.43 for ten computers. In the experiment with stopping criteria t<0.0001 and leapfrog RNG the cost function dropped from 10.77 (sequential processing) to 5.15 for ten computers. For the experiments with length chain of unaccepted changes 1000 iterations the cost function improved from 11.15 (sequential processing) to 5.15 for 10 computers.

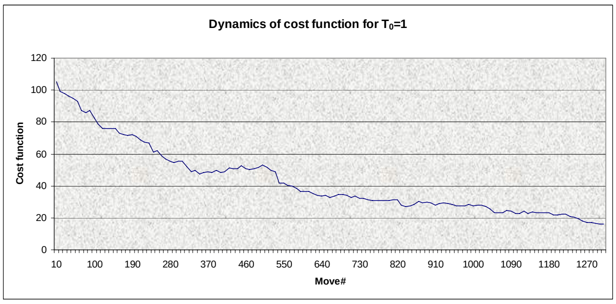

Fig.5. Cost function dynamics for room assignment of 60 persons with initial temperature value $T_o = 1$

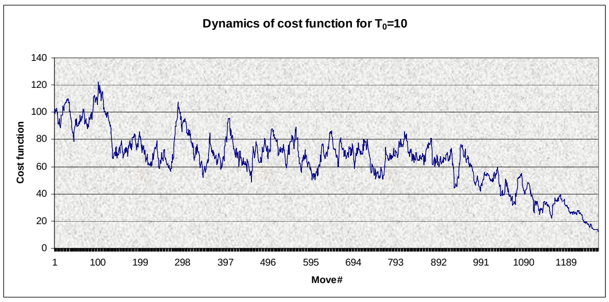

Fig.6. Cost function dynamics for room assignment of 60 persons with initial temperature value $T_o = 10$

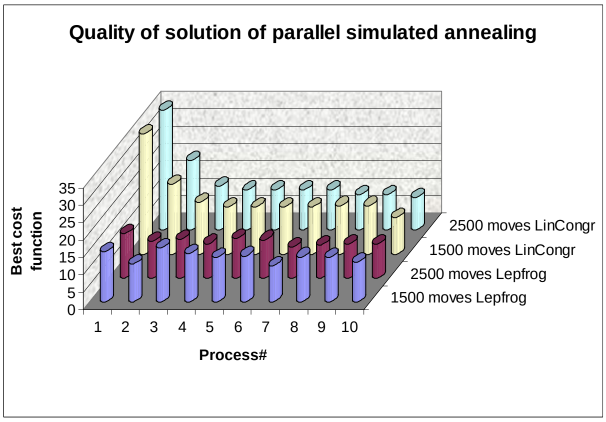

Fig.7. Impact of multicomputer size, the PRNG used and the number of iterations on the quality of solution for room assignment of 60 persons

4 Conclusion

The paper investigates the speedup and quality of solution of parallel simulated annealing applied for the room assignment problem on multicomputer platform. The experimental study is based on multiple independent runs on multicomputer platform. The final value of the cost function is estimated as the global minimum of the best values of the cost function obtained on the different computers. The parallel computational model is based on the parallel programming paradigm "manager-workers".

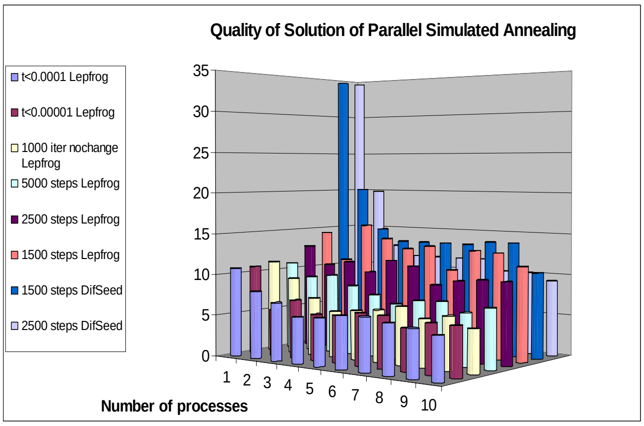

Fig.8. Impact of the termination criteria and the size of multicomputer on the quality of solution for room assignment of 60 persons

Numerous experiments of parallel simulated annealing for solving the room assignment problem have been performed using different parallel random number generators as well as different termination criteria. Performance analysis of the speedup of the parallel system shows that parallel simulated annealing implies good scalability with respect to the size of parallel machine and the size of the parallel application and results in almost proportional increase of the speedup. The analysis of the influence of different parameters on the quality of the solution found in terms of best cost function shows that it is recommended to use parallel leapfrog random number generators because of the less correlation of number sequences and the stopping criteria should be based on the parallel algorithm convergence (minimum temperature or chain length of unaccepted changes) instead of number of iterations.

References:

[1] Metropolis N., A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of State Calculations by Fast Computing Machines, *Journal of Chemical Physics*, Vol.21, pp. 1087 ÷ 1092, 1953.

[2] Aarts E., J. Korst, P. van Laarhoven, Simulated Annealing, in Aarts E., J. Lenstra, eds., *Local Search in Combinatorial Optimization*, John Wiley and Sons, 1997.

[3] Johnson D., C. Aragon, L. McGeoch, Optimization by Simulated Annealing: An Experimental Evaluation; Part I: Graph Partitioning, *Operations Research*, Vol.37, No.6, pp.865÷892, 1989.

[4] Johnson D., C. Aragon, L. McGeoch, Optimization by Simulated Annealing: An Experimental Evaluation; Part II: Graph Coloring and Number Partitioning, *Operations Research*, Vol.39, No.3, pp.378÷406, 1991.

[5] Jeong C., M. Kim, Fast Parallel Simulated Annealing For Traveling Salesman Problem on SIMD Machines with Linear Interconnections, *Journal of Parallel Computing,* Vol.17, pp.221÷228, 1991.

[6] Din D., S. Tseng, A Simulated Annealing Algorithm for Extended Cell Assignment Problem in a Wireless ATM Network, Proc. of EvoWorkshops 2001: *Applications of Evolutionary Computing*, Como, Italy, pp. 150÷160, 2001.

[7] Elmohamed S., G. Fox, P. Coddington, A Comparison of Annealing Techniques for Academic Course Scheduling, *Proc. of 2nd International Conference on the Practice and Theory of Automated Timetabling*, Syracuse, NY, USA, pp. 146÷166, 1998.

[8] Abdennadher S., M. Marte, University course timetabling using constraint handling rules. *Journal of Applied Artifficial Intelligence*, Vol.14, no.4, pp.311÷326, 2000.

[9] Wilke P., M. Gröbner, A Hybrid Genetic Algorithm for School Timetabling, *Proc. of 15th Australian Joint Conference on Artificial Intelligence*, Canberra, Australia, pp.455÷464, 2002.

[10] Salamon P., P. Sibani, R. Frost, Facts, Conjectures and Improvements for Simulated Annealing, *Society for Industrial and Applied Mathematic Monographs on Mathematical Modeling and Computation*, SIAM, Philadelphia, USA, 2002.

[11] Martinez-Alfaro H., J. Minero, G. Alanis, N. Leal, I. Avila, Using Simulated Annealing to Solve the Classroom Assignment Problem, *Proc. of ISAI/IFIS Int. Conference on Intelligent Systems Technologies*, pp.370÷377, 1996.

[12] Osman I., Heuristics for the Generalized Assignment Problem: Simulated Annealing and Tabu Search Approaches, *Journal Operational Research Spectrum*, Vol.17, No.4, pp.211÷225, 1995.