HYBRID PARALLEL SIMULATED ANNEALING USING GENETIC OPERATIONS

Miloš Ohlídal(1)
Josef Schwarz(2)

(1), (2)Brno University of Technology, Faculty of Information Technology
Department of Computer Systems, Božetěchova 2, 612 66 Brno, CZ
Tel.: +420-5-41141210, fax: +420-5-41141270, e-mail: {ohlidal, schwarz}@fit.vutbr.cz

Abstract: *This paper deals with a new algorithm of a parallel simulated annealing HGSA which includes genetic crossover operations. The genetic crossover is used as an enhancement of the origin parallel simulated annealing PSA which allows to recombine solutions produced by individual simulate annealing processes at fixed time intervals. It is found that the proposed algorithm can speed—up the search the global optimum more effectively, compared to PAGASA [1] algorithm and parallel simulated annealing PSA. The performance of the HSGA algorithm is tested on the three known TSP benchmark.*

Key words: *Optimization Problems, Parallel Simulated Annealing, Genetic Crossover, Hybrid Algorithm*

1 Introduction

Simulated Annealing (SA) is one of the frequently used algorithms known as an effective technique for solving combinatorial optimization problems [2]. SA is based on the analogy to the solid annealing and it simulates the process of hardening the solid from the high temperature state to the equilibrium state [3]. Starting with an initial solution (vector of variables) obtained by random or constructive means, the annealing algorithm is a sequence of small random perturbations. The perturbation that improves solution is always accepted, whereas a perturbation that worsens the current solution by an amount ΔC, based on predefined cost function, is accepted with probability $e^{-\frac{\Delta C}{kT}}$, where T is control parameter analogous to the temperature in the annealing of physical system [2][4]. SA can find global solution, however requires huge computational time. There are two solutions of this problem: Parallelization SA or aggregation SA with other optimization algorithms.

Generally, GA parallelization is very simple whereas the SA is naturally sequential. On the other hand it is mathematically proved, that SA converges steadily to the solution. However such a strong evidence does not hold for GA. Therefore, the hybrid method of SA with GA operators is good approach for parallelization of the optimization process.

In this paper, it is proposed parallel simulated annealing using genetic algorithm. This algorithm is a hybrid SA using the GA operations. The proposed algorithm can reduce the computational cost even in continuous problems. The performance of the designed algorithm is tested on the known TSP benchmarks and the effectiveness of the proposed algorithm will be discussed in this paper.

1.1 Parameter setting

- Initial temperature T_0: It must be chosen so that almost all perturbations are accepted.

  y=(number of perturbations accepted) / (total number of perturbations attempted)

$$T = \frac{(\overline{\Delta Cost}^{+})}{\ln(\frac{m^{+}}{y(m^{-}+m^{+})-m^{-}})}, \tag{1.1}$$

  where y is the acceptance probability, $\overline{\Delta Cost}^{+}$ is the average change in cost over all perturbations, which lessen cost function, m^- is the number of perturbations with the cost function decrease and m^+ is the number of perturbations with the cost function increase.

- k_{max}: number of iterations of Metropolis algorithm in one temperature phase. The number k_{max} is based on the requirement that at each value of T quasi-equilibrium is succeeded.

$$k_{\max} = \max_{X_i \in \Omega} |N(X_i)| \tag{1.2}$$

where $N(X_i)$ is size of the vector subspace.

- Decrement coefficient α: The coefficient α (the term in brackets) is proposed to reduce the temperature

$$T_{k+1} = T_k \left\{ 1 + \frac{T_k \ln(1+\delta)}{3\sigma_{T_k}} \right\}^{-1}, \tag{1.3}$$

where δ is a measure of how close the equilibrium vectors of two successive iterations are to each other, σ_{T_k} is the standard deviation of the cost function up to the temperature T_k. The stopping criterion is based on the monitoring of the relevant reduction of the cost function during the optimisation process

$$\frac{d\overline{Cost_s}(T)}{dT} \frac{T}{\overline{C}(T_o)} < \varepsilon_s, \tag{1.4}$$

where ε_s is a small positive number called the stopping parameter, $\overline{C}(T_o)$ is the average value of the cost function at T_0. This condition is based on extrapolation of the smoothed average cost $\overline{Cost_s}(T)$ obtained during the optimization process.

This is a theory how to set the SA parameters. But in practice the value of SA parameters of some problems are known or are determined experimentally. This is the case of TSP problem solved in this paper

1.2 A short survey of hybrid parallel genetic simulated annealing

There are many hybrid parallel genetic simulated annealing algorithms (HGSA), but there are only two main concepts. The first one is based on the algorithm SA which is enhanced with particular genetic operations. The second one is based on the concept of GA which uses Metropolis algorithm at the selection process. In the paper we analyzed three variants of HGSA:

- S. W. Mahfoud and D. E. Goldberg proposed algorithm based on the concept of GA which uses the Metropolis algorithm in the selection process [5].
- M. Krajíc described parallel hybrid genetic simulated annealing, which is based on concept of SA and it uses genetic operations (mutation and crossover)[1].
- N. Mori, J. Yoshida and H. Kita suggested the thermodynamical selection rule in genetic algorithm [6].

We proposed the hybrid parallel genetic simulated annealing using architecture master-slave. Each processes including master process execute simple SA algorithm. The crossover and mutation operations are used just after the communications between processes. The detail description is presented in the next chapters.
The paper is organized as follows: In the second chapter the parallel simulated annealing is described. In the third chapter the structure of HGSA is analyzed more in details. The experimental results are presented in chapter 4.

2 Parallel Simulated Annealing (PSA)

2.1 Simulated annealing

Simulated Annealing (SA) algorithm has basically three important phases: solution generation, acceptance criterion and cooling. SA represents a one-point solution generation. The new point/solution is created by a small perturbation. The acceptance criterion then judges if the transition from the current solution to the new solution is realized. This acceptance criterion is defined as follows:

$$\texttt{random()} < \min[1, e^{-\frac{f(x')-f(x)}{T}}],$$

where $\boldsymbol{x}$ is the vector of variables, $f(\boldsymbol{x}')$ is the new cost function, $f(\boldsymbol{x})$ is old cost function, T is temperature and random() is an random number.
According to this criterion, even when the value of the next solution is worse, the solution can be accepted. In the cooling phase the new temperature is determined by the decrement function $T_{i+1}=\alpha T_i$ [2], where T_i is the i-th temperature stage and α determines the gradient of cooling. The three operations- generation, acceptance criterion and cooling are repeated until the termination condition is reached.

2.2 Parallel simulated annealing

Parallel simulated annealing is based on mutual cooperation of the master-slave processes. These PSA algorithms [7] are divided in two phase. In the first phase all processes are independent and each of them produces its optimised solution. The processes are independent because the communication could be too frequent at higher temperature and the time of communication could be much greater than the time of the optimization process. In the second phase processes cooperate by using the architecture master – slave and all slaves (and also master) work on its sequence of solutions. Generally, if some slave process accepts a solution, it sends it to master, which determines according its own rule of acceptance. If this solution is accepted or some new solution found by master itself, it is sent to all slave processes. In our case, the communication between master and slaves is performed after defined number of iterations at each temperature phase of Metropolis algorithm (e.g. after each 100th or 1000th iteration) and at the end of temperature phase.

The general principle of the scheduling of the messages are shown in the Fig.1.

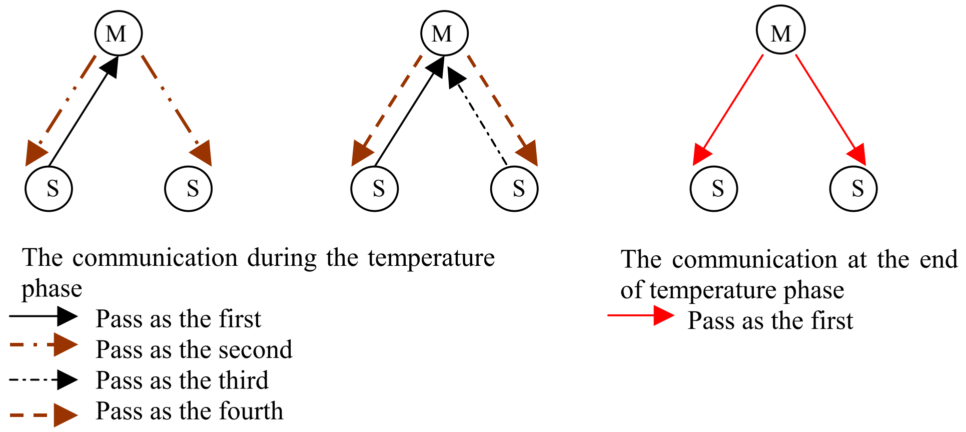

Fig.1: Illustration of the communication during the temperature phase and at the end of the temperature phase

3 Hybrid parallel genetic simulated annealing (HGSA)

In this paper, we proposed parallel simulated annealing using genetic operations. HGSA is a hybrid method and it uses parallel SA with the operations that are used in genetic algorithms. The flow of the evolution is shown in Fig. 2 – this figure was published in [8]. In the proposed algorithm, there are parallel processes and the sequential SA is running in each process. After some steps (after each 1000th iterations of Metropolis algorithm), the crossover is used to produced new solution. The performance of HGSA was tested on the traveling salesman problem.

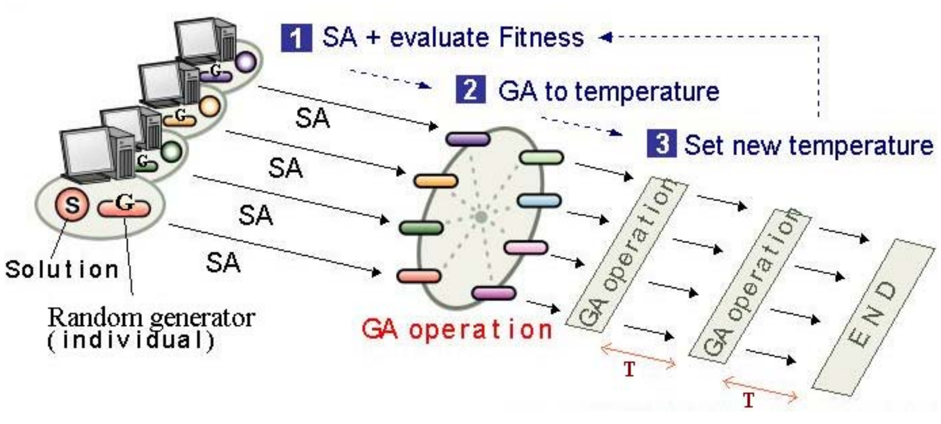

Fig.2: Hybrid parallel genetic simulated annealing

During communication, which is activated each 1000th iteration of Metropolis algorithm, each process sends its solution to master. Master keeps to oneself one solution and one random chosen solution is sent to each slave. These activities are based on the roulette wheel, where the biggest probability of selection has the individual with the smallest length of tour.
After communication all processes have two individuals. Now the phase of genetic crossover starts. From two parent solution two children solution are generated. Because we solved the TSP problem, we used partially matched crossover PMX which produces only feasible solutions.

After crossover there are two parents and two children solutions. Then solution with the smallest tour length is selected and mutation is performed. In case of parent solution mutation is always performed, otherwise the mutation is performed by predefined probability. The mutation is realized by pairwise interchange of cities in the tour/solution -

random generated city is interchanged with his left neighbouring city. A new solution is selected from actual solution of SA process and from the solution, which was obtained after genetic mutation. It is selected using the Metropolis criterion.

3.1 Control parameters of PSA

T_{max}	initial/maximum temperature
T_{min}	final/minimum temperature
α	gradient of cooling ($T= \alpha*T$)
k_{max}	count of iterations of Metropolis algorithm in one temperature phase
Pmut	mutation probability of offspring
Iter	iteration of Metropolis algorithm, where the processes communication each other

4 Experimental results

Hybrid genetic simulated annealing and variants of PSA algorithm were tested on three TSP problems, which were published on the website [9]. The most tests are performed on the benchmark of 52 cities see Fig.3 to 5. It was performed 15 runs for 52 cities in each versions of PSA. The efficiency of PSA versions were also proved by benchmark of 79 cities see Fig.6 and by benchmark of 100 cities.

Optimal solution of TSP problems:

- berlin52 - TSP52 (52 cities) - tour length equals to 7542
- eil79 - TSP79 (79 cities) - tour length equals to is 538
- kroA100 – TSP100 (100 cities) - tour length equals to is 21282

In all experiments the following control parameters were used:

K_{max}	10000
T_{max}	100
T_{min}	1
Alpha	0,9
Prmut	0,1
Iter	1000
Number of processors	6

Tab.1: The value of SA parameters

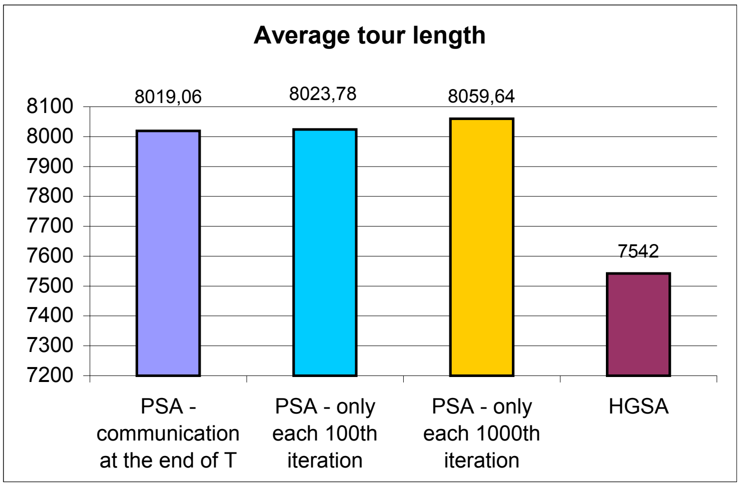

Fig.3: Average tour length of TSP 52 for HGSA and three versions of PSA

In Fig. 3 there are shown experimental results of three PSA versions. The versions differ only by the used time interval between master-slave communication. HGSA algorithm has fixed time interval – communication is performed after each 1000^{th} iterations. All PSA versions found similar average tour length. The optimal tour was not found in any of 15 runs. But in case of HGSA the optimal solution was achieved at each of 15 runs.

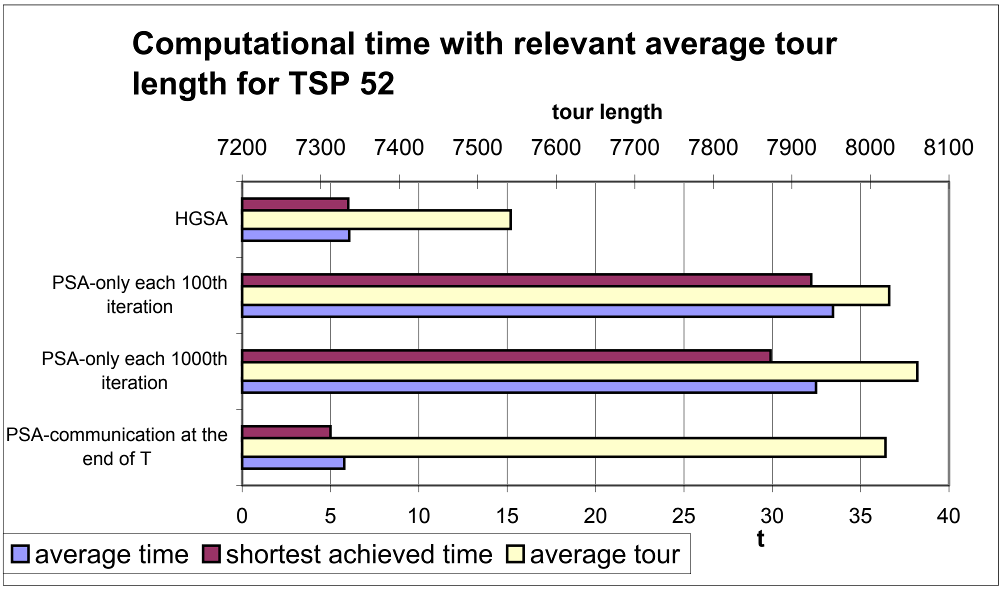

Fig.4: Computational time with relevant average tour length for HGSA and PSA versions

In Fig. 4 the computational time and average tour length is shown. It is evident that the best solution provides HGSA and its computation time is equal to the fastest PSA version.

In Fig.5 and Fig.6 the optimization curves of tour length are presented for HGSA and three PSA versions. It is displayed only the optimization process from temperature phase T=30 but the T_{max} equals to 100. The optimilization process runs all the computation time (from T_{max} to T_{min}).

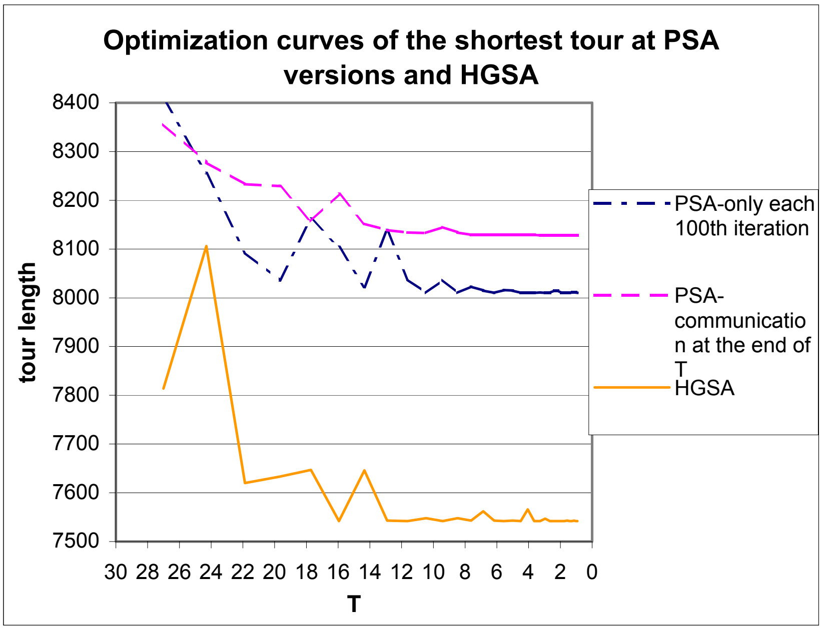

Fig.5: Optimization process for TSP 52 (52 cities)

In Fig.6 optimization curves of three PSA algorithms and the HGSA are presented for TSP 79 problem. It was performed only five experiments. The only HGSA achieves the global minimum.

An extra experiment was applied for comparison the performance of HGSA with PAGASA [1] on the TSP benchmark of 100 cities. HGSA achieves the tour length equals to 21295 which is better than the tour length 21443 achieved by PAGASA. Using the benchmark of 79 cities, PAGASA algorithm gets the best tour length of 540 units and HGSA achieved 538 unit tour length, which is the global optimum.

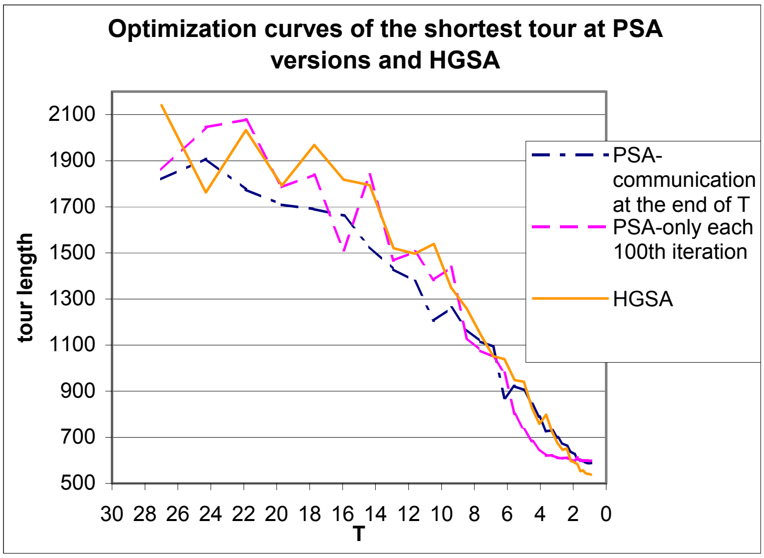

Fig.6: Optimization process of tour length for TSP 79 (79 cities)

5 Conclusions

We have developed a new hybrid optimization algorithm HSGA as an aggregation of parallel simulated annealing PSA and genetic algorithm. We have tested HGSA on three benchmarks of the traveling salesman problems: TSP52, TSP79 and TSP100. The comparison of the performance of HSGA and PSA was realized. HSGA algorithm achieved the global optimum in each of 15 runs for TSP52. The PSA version received only a local solution. Another experiment was arranged as a comparison of HSGA and a version of hybrid PSA called PAGASA published in [1]. Four 100 cities TSP benchmark HSGA outperforms PAGASA. The future works will be focused on further testing of HSGA on the most complex benchmarks and development of advanced version of HSGA algorithm.

Acknowledgement

This research has been carried out under the financial support of the Research intention no. CEZ: J22/98: 262200012- "Research in information and control systems" (Ministry of Education, CZ) and the research grant GA 102/02/0503 "Parallel system performance prediction and tuning" (Grant Agency of Czech Republic).

References

[1] Krajíc, M.: *Algorithm of parallel hybrid genetic simulated annealing to solve of traveling salesman problem*, ČVUT FEL, Prag, 2002

[2] Rosen, B. E. and Nakano, R.: Simulated annealing -basics and recent topics on simulated annealing. *Proceeding of Japanese Society for Arti-ficial Intelligence*, Vol. 9, No. 3, 1994.

[3] Aarts, E. H. L. and Korst, J. H. M.: *Simulated Annealing and Boltzmann Machines*. John Wiley Sons, 1989.

[4] Kita, H.: Simulated annealing. *Proceeding of Japan Society for Fuzzy Theory and Systems*, Vol. 9, No. 6, 1997.

[5] Mahfoud, S. W. and Goldberg, D. E.: A genetic algorithm for parallel simulated annealing. *Parallel Problem Solving from Nature 2*, pp. 301–310, 1992.

[6] Mori, N., Yoshida, J. and Kita, H.: Suggestion of thermodynamical selection rule in genetic algorithm Transaction of Institute of Systems, Control and Information Engineers, Vol. 9, No. 2, pp. 82–90, 1996.

[7] Ohlídal, M.: Programming of versions of parallel simulated annealing algorithm PSA *and their testing*, in Czech,VUT FIT, Brno, 2003

[8] Miki, M., Hiroyasu, T., Wako, J., Yoshida, T.: *Adaptive Temperature Schedule Determined by Genetic Algorithm for Parallel Simulated Annealing*, Doshisha University, Kyoto,2003

[9] MP – TESTDATA
URL: *http://elib.zib.de/pub/mp-testdata/tsp/tsplib/index.html*

[10] Manual MPI. Document reasonable on URL:
http://bitscap.bits-pilani.ac.in/param/public_html/mpi_basic_calls.html, (May 2002)