Detailed Analysis of Uphill Moves in Temperature Parallel Simulated Annealing and Enhancement of Exchange Probabilities

Yang Jun
School of Information Science,
Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Tatsunokuchi, Ishikawa 923-1292, Japan

Satoshi Mizuta
Faculty of Science and Technology,
Hirosaki University,
3 Bunkyo-cho, Hirosaki-city, Aomori 036-8561, Japan

This paper demonstrates that the probability of the occurrence of uphill moves in temperature parallel simulated annealing is so small that the effects of parallelization might be lost when the number of processing nodes is small. A modification in the probability of solution exchange between the processes is one tactic to recover the parallelization effects. This method is evaluated by a traveling salesman problem with 33 cities in China. The results of the experiments show that, although it violates the detailed balance condition, the modification can improve the quality of the solutions.

1. Introduction

In many fields of industry and technology, solving combinatorial optimization problems is an important process to increase productivity and product quality. The objective of solving such problems is to search for the optimal solution, that is, the solution with the lowest cost or the highest profit out of many feasible solutions. Usually, it is impossible to find the global optimal or a near-optimal solution in practical time by checking all the feasible solutions because the combinatorial optimization problems tend to belong to NP-hard or NP-complete problems, which have solution spaces growing exponentially with the size of the problem. Therefore, metaheuristic approaches, which can deliver a highly effective solution in polynomial execution time, are usually used to solve combinatorial optimization problems although it is not guaranteed that the available solution is the global optimal.

One of those algorithms is simulated annealing (SA) [1], in which a Monte Carlo (MC) simulation is performed while a control parameter called *temperature* is gradually being changed. However, the execution

https://doi.org/10.25088/ComplexSystems.15.4.349

time required to get a solution with good quality is still long in practice. One approach to reduce the execution time is by using probability distributions other than the conventional Boltzmann distribution [2, 3] to control the appearance of states. Parallel implementation of SA (PSA) is another trend that is being investigated energetically in these days because high-performance PC clusters are getting more available.

PSAs are classified into two categories: problem-dependent and problem-independent approaches. In the former, parallelism is achieved according to the features of the search space or the objective function. This approach is, therefore, suitable for some special problems. In the latter, on the other hand, parallelism is achieved by decomposing the algorithms and employing parallel moves. Because the method in this approach parallelizes SA itself, it can be applied to any optimization problem, and much research for methods based on this approach has been carried out [4, 5, 6].

Temperature parallel simulated annealing (TPSA) [7, 8] is one of the problem-independent approaches, in which MC simulations are executed on multiple processors independently at the individually assigned different temperatures. It is distinguished from other PSAs on the point that each MC simulation is performed at the fixed temperature. Strictly speaking, therefore, it is not an annealing, but as a new paradigm of PSA it bas been studied by several researchers. For instance, Miki *et al.* introduced the adaptive neighborhood into TPSA [9] and investigated the important temperature in TPSA [10].

In this paper, we introduce another method to improve TPSA, which is thought to be easily implemented. The remaining portion of this paper is organized as follows. In section 2, SA and TPSA are briefly reviewed. In section 3, an uphill move in TPSA is defined and analyzed in detail. Moreover, a modification in the solution exchange probability is introduced. The procedure of the experiments and their results are illustrated in section 4. Then, summary and discussion are included in section 5.

2. SA and TPSA

In this section, we briefly review SA and TPSA.

SA [1] is a randomized local search method based on the simulation of annealing a metal. The procedure is represented in pseudocode format in Figure 1.

The acceptance probability of a trial solution is given by

$$p_{\mathrm{ac}}(T, \Delta E) = \begin{cases} 1 & \text{if } \Delta E < 0 \\ \exp\left(-\frac{\Delta E}{T}\right) & \text{otherwise,} \end{cases} \tag{1}$$

where T is the *temperature* of the system, and ΔE is the difference of the costs between the trial and the current solutions, E' and E, re-

```
Begin
choose the initial solution;
calculate the cost;
Repeat ... outer loop
    Repeat ... inner loop
        perturb the current solution;
        calculate the cost;
        accept the trial solution as a new solution by acceptance probability;
    Until iteration limit is reached;
temperature is lowered according to the cooling schedule;
Until termination condition is satisfied;
End
```

Figure 1. Pseudocode representation of SA.

spectively ($\Delta E \equiv E' - E$). Equation (1) means that the trial solution is accepted by a nonzero probability $\exp(-\Delta E/T)$ even though the solution deteriorates. In this case, because the cost of the solution increases, we call this update on the solution an *uphill move*.

Uphill moves enable the system to escape from the local minima; without them, the system would be trapped into a local minimum. Too high of a probability for the occurrence of uphill moves, however, prevents the system from converging. In SA, the probability is controlled by temperature in such a manner that at the beginning of the procedure the temperature is sufficiently high, in which a high probability is available, and as the calculation proceeds the temperature is gradually decreased, lowering the probability.

The procedure of SA consists of two nested loops: an *outer* loop and an *inner* one (see Figure 1), where the cooling and the equilibrating processes are performed, respectively. TPSA [7, 8] is a parallel measure of the outer loop, in which multiple processes independently perform the equilibrating loop at each temperature chosen by a rule corresponding to the cooling schedule for the sequential counterpart. At a certain interval of the execution, the processes having adjacent temperatures exchange their current solutions by the following exchange probability,

$$p_{\text{ex}}(T_i, T_{i+1}, \Delta E_i) = \begin{cases} 1 & \text{if } \Delta E_i \cdot \Delta T_i < 0 \\ \exp\left(-\frac{\Delta E_i \cdot \Delta T_i}{T_i \cdot T_{i+1}}\right) & \text{otherwise,} \end{cases} \tag{2}$$

where T_i is the temperature of process i, and ΔT_i is the difference between the adjacent temperatures, T_i and T_{i+1} ($\Delta T_i \equiv T_i - T_{i+1}$); ΔE_i is the difference of the costs between processes i and $(i+1)$, E_i and E_{i+1}, respectively ($\Delta E_i \equiv E_i - E_{i+1}$).

Figure 2 depicts the behavior of TPSA in the case of four processes, where the vertical dotted arrows show the points of solution exchanges. After enough iterations, the solution on the process having the lowest

https://doi.org/10.25088/ComplexSystems.15.4.349

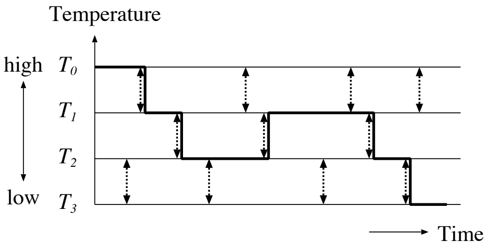

Figure 2. Behavior of TPSA in the case of four processes. The vertical dotted arrows and the meandering thick line indicate the solution exchanges and the flow of the finally accepted solution, respectively.

temperature is expected to converge to the global optimal. The meandering thick line in Figure 2 shows the flow of the solution that is finally accepted.

Note that the exchange probability of equation (2) satisfies the detailed balance condition,

$$\begin{aligned} &\mathrm{e}^{-E_i/T_i} \cdot \mathrm{e}^{-E_{i+1}/T_{i+1}} \cdot p_{\mathrm{ex}}\,(T_i, T_{i+1}, \Delta E_i) \\ &\qquad = \mathrm{e}^{-E_{i+1}/T_i} \cdot \mathrm{e}^{-E_i/T_{i+1}} \cdot p_{\mathrm{ex}}\,(T_i, T_{i+1}, -\Delta E_i). \end{aligned} \tag{3}$$

3. Detailed analysis of uphill moves in TPSA

Uphill moves are considered to play a significant role in searching for the global optimal in TPSA as well as in conventional SA. Therefore, the efficiency of uphill move operations affects the performance of TPSA. Here, we first define uphill moves in TPSA and analyze them in detail. Next, we introduce a modification in the solution exchange probability in order to make the uphill moves in TPSA more effective.

There are two kinds of uphill moves in TPSA, *intra-* and *interprocess*. The intraprocess uphill moves arise within the equilibrating loop on each process and have the same effects as those in SA. On the other hand, the interprocess uphill moves are peculiar to TPSA and arise through the solution exchange processes. Here, we concentrate our discussion on interprocess uphill moves.

Usually, the relationship between the adjacent two temperatures is chosen as

$$T_{i+1} = \alpha T_i, \qquad 0 < \alpha < 1. \tag{4}$$

Here, we define an uphill move in TPSA as the exchange of the current solutions satisfying $E_i > E_{i+1}$ because the cost of the solution on the process with the lower temperature T_{i+1} increases from E_{i+1} to E_i after the solution exchange. The probability of the occurrence of the uphill move is calculated from equations (2) and (4) as

$$p_{\text{up}} = \exp\left(-\frac{\Delta E_i \cdot \Delta T_i}{T_i \cdot T_{i+1}}\right) = \exp\left[-\frac{\Delta E_i}{T_{i+1}} \cdot (1-\alpha)\right]. \tag{5}$$

On an equilibrium state, the cost typically has the order of magnitude of the system's temperature. Thus, we can roughly estimate the costs of the solutions on processes i and $(i+1)$ as $E_i \sim T_i$ and $E_{i+1} \sim T_{i+1}$, respectively. From equation (4), we can get

$$\Delta E_i \sim T_i - T_{i+1} = (1-\alpha)T_i = \frac{(1-\alpha)}{\alpha}T_{i+1}, \tag{6}$$

and equation (5) becomes

$$p_{\text{up}} = \exp\left[-\frac{(1-\alpha)^2}{\alpha}\right]. \tag{7}$$

Next, let us estimate α. From equation (4), we can derive $\alpha = (T_{N-1}/T_0)^{1/(N-1)}$, where N is the number of the available processes. The highest and lowest temperatures, T_0 and T_{N-1}, respectively, are usually fixed according to the problem dealt with. We treat, therefore, $t \equiv T_{N-1}/T_0$ as a constant. Substituting these formulas into equation (7), we can express p_{up} as a function of N:

$$p_{\text{up}} = \exp\left(-t^{1/(N-1)} - t^{-1/(N-1)} + 2\right). \tag{8}$$

Figure 3 shows p_{up} calculated by equation (8) for $t = 10^{-2}$, 10^{-3}, 10^{-4}, and 10^{-5}. Generally speaking, if the lowest temperature is not sufficiently low, the system does not converge. On the other hand, if the highest temperature is not sufficiently high, the system tends to fall into a local minimum. Therefore, t must be sufficiently small; for example, $t = 10^{-5}$.

Notice in Figure 3 that p_{up} is nearly zero when the processes are fewer than eight in the case of $t = 10^{-5}$. In that case, uphill moves almost never occur, and the system is apt to be trapped into a local minimum. One of the tactics to avoid this difficulty is assigning multiple processes to each processing node, where we can totally execute many processes even if enough nodes are not available. However, because the processes assigned to each single node are executed sequentially, the additional overhead, such as interprocess communications, may make the TPSA worse than the conventional annealing counterpart. Alternatively, we modify the exchange probability (equation (2)) so that it becomes adequately large for small N.

In order to increase the probability for small N, we introduce a parameter n into equation (2) for the case of $\Delta E_i \cdot \Delta T_i \leq 0$ as

$$\exp\left(-\frac{\Delta E_i \cdot \Delta T_i}{T_i \cdot T_{i+1}}\,\alpha^n\right), \tag{9}$$

https://doi.org/10.25088/ComplexSystems.15.4.349

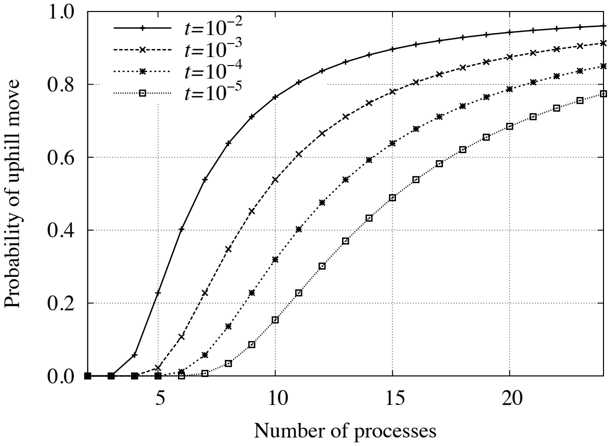

Figure 3. Probability of the occurrence of uphill moves in TPSA using equation (8). t is defined as $t \equiv T_{N-1}/T_0$, the ratio of the lowest temperature to the highest one.

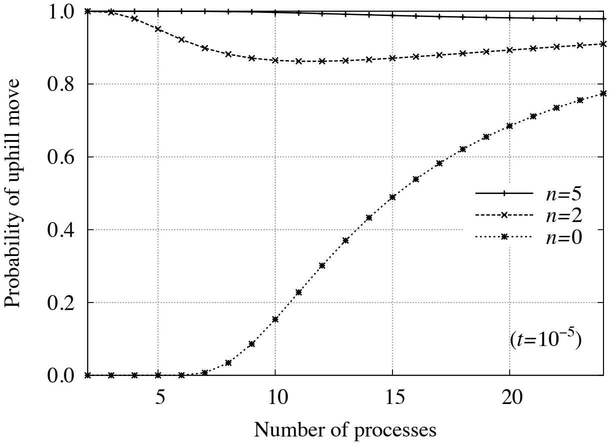

Figure 4. Modified probability of the occurrence of uphill moves ($t = 10^{-5}$).

which modifies equation (8) to

$$p_{\text{up}} = \exp\left(-t^{(n+1)/(N-1)} - t^{(n-1)/(N-1)} + 2t^{n/(N-1)}\right). \tag{10}$$

Note that this modification shows its effects only for small N and that the original formula recovers for $n = 0$.

Figure 4 shows p_{up} (equation (10)) *versus* the number of processes for $n = 0, 2$, and 5 (t is fixed to 10^{-5}). Note that the probability remains large even in the small N region.

4. Experiments and the results

We evaluated the effects of the modified probability (equation (9)) by applying it to solve a traveling salesman problem (TSP). We chose 33 cities that have the province governments in China. Figure 5 shows their locations and the global optimal tour calculated by *Concorde* [11] (the length of the optimal tour is 16,706 km).

The highest temperature T_0 is set to 2164 so that the acceptance probability (equation (1)) becomes 0.5 for $\Delta E = 1500$, and the lowest temperature T_{N-1} is chosen as 0.1, which yields $t \sim 4.6\times10^{-5}$ ($\alpha \sim 0.19$). A trial solution is generated by exchanging the visiting order of two cities randomly chosen in the current solution.

The experiments are performed on a PC cluster consisting of seven nodes, the number of iterations in the equilibrating loop being changed. Each node has a Pentium III CPU of 600MHz and 256MB RAM, and the nodes are interconnected by a 100Mbps Ethernet LAN through a switching hub. The programs are coded in the C language with LAM-6.5.6 MPI library.

Figure 6 shows the results of the experiments for $n = 0$ (no modification), 2, and 5. The horizontal axis is the number of iterations in the equilibrating loop, and the vertical one is the resultant tour length. The averages of 20 runs and the standard deviations are shown. The number of solution exchanges is set to 100, which means that, for example, the maximum number of iterations is 6×10^6 per process. Because we fixed both the number of solution exchanges and that of iterations in the equilibrating loop for all the processes for each run, the execution time is independent of the number of processes. Therefore, speedup is meaningless here. Actually, the execution time is roughly proportional

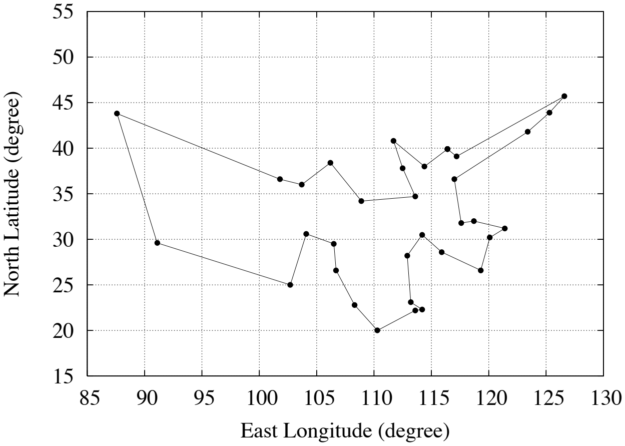

Figure 5. Locations of the cities in China and the optimal tour.

https://doi.org/10.25088/ComplexSystems.15.4.349

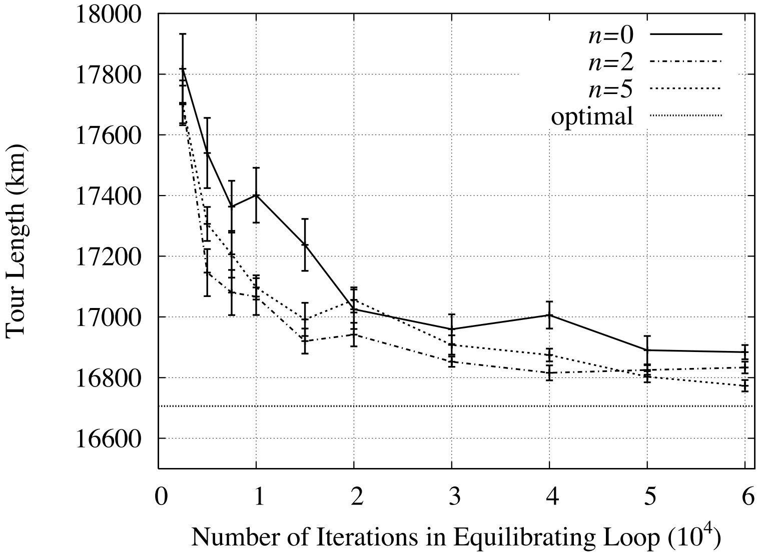

Figure 6. Results of experiments. The averages of 20 runs and the standard deviations are shown.

to the number of iterations in the equilibrating loop; for instance, it is about 10 seconds for 6×10^4. Instead of speedup, we consider the quality of the solutions as the measure of improvement. We can notice in Figure 6 that the solutions are improved by the modification in almost the whole range of the horizontal axis.

In order to investigate the influence of the number of processes, which are equal to the number of processing nodes in this study, on our algorithm, we also performed experiments with four and 14 processes. The results are shown in Figure 7. The effects of the enhancement of the exchange probability are strongly noticed for the case of four processes, although the resultant tour length is far from the optimal because of the lack of processes. On the other hand, little effect is available for the case of 14 processes. These results are consistent with the argument in the previous sections.

5. Summary and discussion

In this paper, we pointed out that the probability of the occurrence of uphill moves in temperature parallel simulated annealing (TPSA) is so small that the effects of the parallelization might be lost when only a few processing nodes are available. In order to overcome the problem, we introduced an enhancement factor into the solution exchange probability.

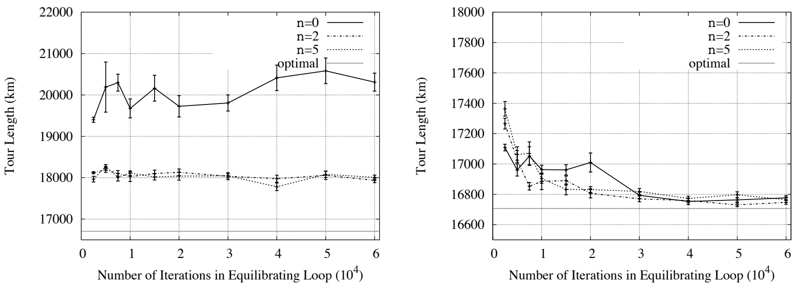

Figure 7. Results of experiments with the different number of processors: the left for 4 and the right for 14. The averages of 20 runs and the standard deviations are shown.

We evaluated the effects of the modification by a traveling salesman problem with 33 cities in China and showed that the modification improved the quality of the resultant solution compared to the unmodified case.

Here, we should mention the detailed balance condition of equation (3). In general, the solution exchange probability is defined so that it satisfies the detailed balance condition. Even though it is one of the sufficient conditions for the system to reach the equilibrium state, it is not necessarily required for getting the optimal or a near-optimal solution with respect to practical cases. In fact, we often choose a rapid cooling schedule in executing the conventional simulated annealing (SA) such as $T \propto s^{-k}$, where s is the number of steps and k is a positive constant, although the convergence to the global optimal is guaranteed only for cooling schedules slower than $T \propto 1/\log s$ [12].

The modified probability (equation (9)) indeed violates the detailed balance condition (equation (3)). Therefore, an equilibrium state achieved on a process is destroyed when a solution exchange occurs on that process. However, because our aim is not to perform the simulation itself exactly but to get the optimal solution in a practical way, this violation is thought to be forgivable as far as it has good effects on the calculation.

Acknowledgments

We thank Professor Toshio Shimizu for reading the manuscript carefully and giving us helpful comments.

References

[1] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science*, **220** (1983) 671–680.

[2] Harold Szu and Ralph Hartley, "Fast Simulated Annealing," *Physics Letters A*, **122** (1987) 157–162.

[3] T. J. P. Penna, "Traveling Salesman Problem and Tsallis Statistics," *Physical Review E*, **51** (1995) 1–3.

[4] Daniel R. Greening, "Parallel Simulated Annealing Techniques," *Physica D*, **42** (1990) 293–306.

[5] H. Chen, N. S. Flann, and D. W. Watson, "Parallel Genetic Simulated Annealing: A Massively Parallel SIMD Algorithm," *IEEE Transactions on Parallel and Distributed Systems*, **9** (1998) 126–136.

[6] Esin Onbaşoğlu and Linet Özdamar, "Parallel Simulated Annealing Algorithms in Global Optimization," *Journal of Global Optimization*, **19** (2001) 27–50.

[7] Kenzo Konishi, Kazuo Taki, and Kouichi Kimura, "Temperature Parallel Simulated Annealing Algorithm and Its Evaluation," *Transactions on Information Processing Society of Japan*, **36** (1995) 797–807.

[8] Koji Hukushima and Koji Nemoto, "Exchange Monte Carlo Method and Application to Spin Glass Simulations," *Journal of the Physical Society of Japan*, **65** (1996) 1604–1608.

[9] Mitsunori Miki, Tomoyuki Hiroyasu, Masayuki Kasai, Keiko Ono, and Takeshi Jitta, "Temperature Parallel Simulated Annealing with Adaptive Neighborhood for Continuous Optimization Problem," in *Computational Intelligence and Applications* (Dynamic Publishers, Inc. Atlanta, GA, USA, 2002) 149–154.

[10] Mitsunori Miki, Tomoyuki Hiroyasu, Takeshi Yoshida, and Toshihiko Fushimi, "Parallel Simulated Annealing with Adaptive Temperature Determined by Genetic Algorithm," Doshisha University, 1-3 Tatara Miyakodani, Kyo-tanabe, Kyoto, 610-0321, Japan (2002).

[11] *Concorde TSP Solver*, http://www.tsp.gatech.edu/concorde.html.

[12] Stuart Geman and Donald Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6** (1984) 721–741.