

LA-UR- 08-5100

Approved for public release;  
distribution is unlimited.

*Title:* Parallel Tempering for the Traveling Salesman Problem

*Author(s):* Richard Wang, Jeffrey Hyman, Allon Percus and Russel  
Caflisch

*Intended for:* Physica D



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# Parallel Tempering for the Traveling Salesman Problem

Richard Wang\*, Jeffrey D. Hyman†, Allon G. Percus‡ and Russel Caflisch§

July 29, 2008

## Abstract

We explore the potential of applying parallel tempering to combinatorial optimization. In particular, we use travelling salesman problem as a test case and compare the simulation results computed by parallel tempering and simulated annealing. We study how different choices of parameters affect the simulation results and give insight into relative performance of the two methods. We found that when parameters are chosen appropriately, both methods yield close approximation to the actual minimum distance. However, parallel tempering yields more consistent accurate results when a series of independent simulation are performed. Our results demonstrate that parallel tempering has great potential of becoming an alternative method to simulated annealing for solving optimization problem.

---

\*Mathematics Department, University of California at Los Angeles, Los Angeles, CA 90095-1555 USA, cmwang@math.ucla.edu

†hymanjd@gmail.com

‡Information Sciences Group, Los Alamos National Laboratory, Los Alamos, NM 87545 and Mathematics Department, UCLA, Los Angeles, CA 90095, percus@ipam.ucla.edu. Research supported by the US Department of Energy under contract DE-AC52-06NA25396 through the Laboratory Directed Research and Development Program at LANL

§Mathematics Department, University of California at Los Angeles, Los Angeles, CA 90095-1555 USA, caflisch@math.ucla.edu. Research supported in part by the NSF through grant DMS-0707557

# 1 Introduction

In this work we explore the potential of applying parallel tempering to combinatorial optimization. Parallel tempering (PT), also called replica exchange or simulated tempering [1, 2], is a Monte Carlo method intended primarily for sampling a probability distribution function with a complex structure. The original version of PT was developed by Swendsen & Wang [1]. In their work, replicas of a system of interest were simulated at a set of different temperatures. Replicas at adjacent temperatures undergo a partial exchange of configuration information. Initially the method was applied to systems in statistical mechanics. Recently, it has been successfully applied more broadly [3], including in engineering, biology and material science. The main usage of the method in the literature is to enhance the sampling of configurations. We will use it to find near-optimal solutions to the traveling salesman problem.

The traveling salesman problem (TSP) is to determine the shortest route (“tour”) starting from a home location (“city”), visiting all other cities exactly once and then returning home. The problem is NP-hard, and so when the number of cities is large, it is computationally infeasible to find the true optimal tour: an approximation algorithm must be deployed. One of the simplest constructive approximation techniques is the nearest-neighbor method. In this method, the tour starts from any city and recursively chooses the nearest city not yet visited. Many iterative improvement heuristics, such as  $k$ -opt [4, 5, 6, 7] and Lin-Kernighan [8] have been developed over the past decades.

The most popular Monte Carlo-based optimization method that has been applied to the TSP is simulated annealing [9, 10]. Based on the Metropolis method [11], simulated annealing (SA) includes a schedule of temperatures and approaches the global minimum when the temperatures decrease gradually. There are extensive studies in the literature of SA applied to the TSP. The survey by Johnson et al. [12] remains one of the most comprehensive treatments of how to implement the algorithm and choose the most appropriate parameters, and so we use it as a benchmark for our analysis. By contrast, we are aware of only one preliminary study [13] that explores parallel tempering for the TSP.

Our main contribution in this paper is demonstrating that a straightforward implementation of PT can outperform the SA benchmark in several crucial respects, offering a simple but powerful alternative. We compare the performance of SA and PT on the traveling salesman problem. We study

how different choices of parameters affect the simulation results and give insight into relative performance. It is known that on random instances with hundreds of cities, under appropriate selection of parameters, SA yields approximations that are roughly 1% above optimal [12]. We find that PT finds approximations that are at least as good and typically more consistent, given roughly equivalent computational resources. Moreover, if computational resources are measured in terms of parallel time, our results suggest that PT considerably improves upon SA's results.

The article is organized as follows. In Section II, we describe our implementations of simulated annealing (SA) and parallel tempering (PT). In Section III, we present simulation results for each method. In Section IV, we compare the two methods. In our conclusion we discuss the implications of these findings.

## 2 Methods

The use of Monte Carlo methods for the TSP is based on an analogy between combinatorial optimization problems and statistical mechanics, described in [9]. For example, configurations in combinatorial optimization problems correspond to states in thermodynamics; the cost of a configuration in optimization corresponds to the energy of a state in thermodynamics. In subsequent sections, we use these terminologies interchangeably when both are clear in the context of our discussion. In the case of the TSP, cost is distance, and a configuration is a tour.

### 2.1 Metropolis Method

Simulated annealing and parallel tempering are both based on the Metropolis method [11], one of the most widely used simulation approaches. Let us recall the main elements of this method. The Metropolis algorithm generates a sequence of states for a system in equilibrium at a certain temperature  $T$ . It is an acceptance-rejection method. At each step, given a current state, the method attempts a trial move to a new state, and then determines whether the trial is accepted or rejected using a formula of acceptance probability. Let  $P$  be a probability density of a state,

$$P(i) \equiv \frac{\exp[-\frac{E(i)}{T}]}{Z}$$

where  $Z \equiv \sum_i \exp[-\frac{E(i)}{T}]$  and  $E(i)$  is the energy at the state  $i$ . The selection of the acceptance probability is motivated by the detailed balance condition:

$$P(o)\pi(o \rightarrow n) = P(n)\pi(n \rightarrow o) \quad (1)$$

where  $P(o)$  is the probability at the old state  $o$  and  $\pi(o \rightarrow n)$  is the transition probability from the old state  $o$  to the new state  $n$ . If we define  $\alpha(o \rightarrow n)$  to be the trial transition probability, define  $acc(o \rightarrow n)$  to be the acceptance probability and assume  $\alpha$  is symmetric, then the condition for the acceptance probability can be derived by the detailed balance condition (1) [14]:

$$\frac{acc(o \rightarrow n)}{acc(n \rightarrow o)} = \frac{P(n)}{P(o)} = \exp(-\frac{\Delta E}{T}) \quad (2)$$

where  $\Delta E = E(n) - E(o)$ . The Metropolis acceptance probability, one of many satisfying the equation (2), is the following:

$$acc(o \rightarrow n) = \begin{cases} \frac{P(n)}{P(o)} = \exp(-\frac{\Delta E}{T}) & \text{if } P(n) < P(o) \\ 1 & \text{if } P(n) > P(o) \end{cases}$$

For implementation, a uniform random variable  $\xi$  on  $[0, 1]$  is sampled to determine whether a trial is accepted or not. If  $\xi < \min(1, \exp(-\frac{\Delta E}{T}))$ , then a trial is accepted; otherwise, a trial is rejected. The Metropolis method uses a single temperature and proceeds in small steps from one configuration to another. The temperature allows uphill moves and therefore gives the particles a chance to get out of a local basin. Downhill moves are always accepted.

The Metropolis method has also been used to find a local minimum for a physical system. The temperature  $T$  controls the size of the uphill move for each step. Larger temperature  $T$  allows greater uphill moves, while smaller  $T$  only permits small uphill moves. When the temperature is very low, the method is essentially a greedy algorithm.

The Metropolis method is not effective when applied to a problem which has a high energy barrier or multiple shallow basins. For such problems, a particle is easily trapped in the basin and therefore not able to move freely within the configuration space.

## 2.2 Simulated Annealing

The process of physical annealing begins with heating metal to a high temperature and holding it there for a certain length of time, and then letting it cool down slowly. This process allows an atom in the metal to achieve minimal internal energy. Motivated by the concept of physical annealing, simulated annealing uses a schedule of temperatures to solve optimization problems. SA runs the Metropolis algorithm using a high temperature in the beginning and reducing the temperature slowly, in the hope of reaching a neighborhood that contains a global minimum.

The results of simulated annealing depend on the cooling schedule, i.e., the choice of temperatures and the number of Metropolis steps at each temperature. The most widely used schedule for SA is exponentially decreasing temperatures:

$$T(i) = T_0 r^i$$

where  $T_0$  is the initial temperature,  $r < 1$  is the cooling rate and  $T(i)$  is the temperature used after the  $i$ th reduction. Let  $N_{total}$  be the total number of steps, and  $N_{length}$  the number of steps before a temperature decrease (called temperature length).  $N_{temperatures} = \lceil \frac{N_{total}}{N_{length}} \rceil - 1$  is equal to the number of temperature decreases during the simulation. The algorithm for SA is the following:

1. For  $i = 0, 1, \dots, N_{temperatures}$
2. Run Metropolis algorithm for  $N_{length}$  steps at the temperature  $T(i)$ , until total number of steps  $N_{total}$  has been reached.

As discussed in the previous section, larger temperatures in a Metropolis simulation result in the acceptance of larger increases in energy. Using a wide range of temperatures allows a simulation to explore the energy landscapes before it relaxes and arrives at the ground state. In the beginning, higher temperatures are used so that the configurations giving greater increases in energy will be accepted. This enables a simulation to cross high energy barriers and hop among shallow energy basins, therefore exploring broad energy landscapes. The lower temperatures are used to achieve configurations with small uphill moves. The simulation then focuses on finding the local minimum energy in a small region.

For the traveling salesman problem, our Metropolis trial move is known as a *2-change*. We select two links in the tour, and create a new TSP tour by

deleting these two links and then reconnecting the tour in the only other way possible. For example, say our instance contains eight cities and the current tour is

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8.$$

If the two links chosen are  $1 \rightarrow 2$  and  $6 \rightarrow 7$ , the new tour connects 1 to 6, reverses the order between 6 and 1, and then connects 2 to 7:

$$1 \rightarrow \mathbf{6} \rightarrow \mathbf{5} \rightarrow \mathbf{4} \rightarrow \mathbf{3} \rightarrow \mathbf{2} \rightarrow 7 \rightarrow 8.$$

We base the details of our SA implementation on the method described in [12], as this is among the most competitive ones to date. Specifically:

- Letting  $n$  be the number of cities, set the initial temperature to be

$$T_0 = \frac{1.5}{n}.$$

This results in an initial acceptance rate of approximately  $1/2$  [12].

- Set the cooling rate to be  $r = 0.95$ .
- Use the nearest-neighbor construction heuristic to establish the initial tour: starting from the first city, keep connecting to the nearest city that has not yet been used on the tour, and finally return to the starting city. For Euclidean TSP instances, these initial tours are typically about 20-25% longer than optimal.
- For the trial move, select the two links as follows. Pick a city  $t_1$  at random, and pick a second city  $t_2$  from within a given neighborhood of the first one. The neighborhood, defined in the next section, is such that each city typically has around 20 neighbors. Now pick, with equal probability, one of the two cities connected to  $t_1$ , and call it  $t_3$ . Finally, consider the branch of the tour that goes from  $t_2$  to  $t_3$  *without* passing through  $t_1$ , and let  $t_4$  be the city connected to  $t_2$  on this branch. The 2-change consists of deleting links  $t_3 \rightarrow t_1$  and  $t_4 \rightarrow t_2$ , replacing them by  $t_4 \rightarrow t_1$  and  $t_2 \rightarrow t_3$ .
- Take the temperature length be proportional to the number of possible configurations accessible via a 2-change. Since there are  $n$  possibilities for  $t_1$ , roughly 20 possibilities for  $t_2$  and 2 possibilities for  $t_3$ , let  $N_{length} = 40\alpha n$ . The studies in [12] use values of the proportionality constant *alpha* between 10 and 100.

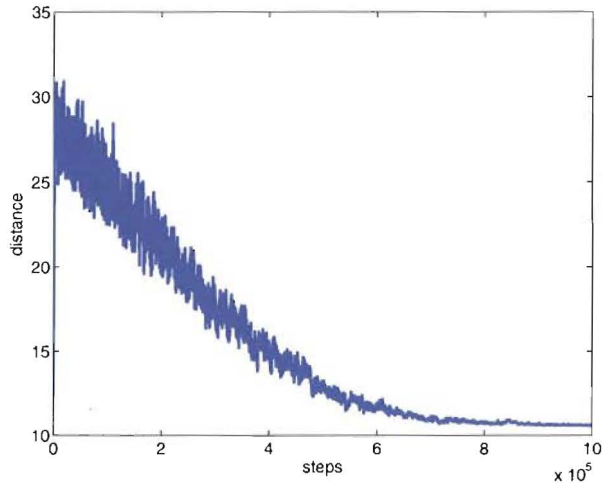


Figure 1: SA with total number of steps  $N_{total} = 1,000,000$ , initial temperature  $T_0 = 1.5/\sqrt{200} \approx 0.1061$ , cooling rate  $r = 0.95$ , and number of steps at each temperature  $N_{length} = 16,000$  (corresponding to  $\alpha = 2$ ).

Figure 1 shows a sample run of SA on a TSP instance with  $n = 200$ . The tour length at first grows significantly from its starting value, and fluctuates greatly, since a large fraction of uphill moves are accepted. As the temperatures decrease, the tour distances also decrease, and the fluctuations gradually diminish.

The efficiency and accuracy of a simulation is largely determined by the cooling schedule; i.e., the temperatures being used and the number of Metropolis steps being run between two successive temperatures. We will discuss how the choice of  $N_{total}$  and  $N_{length}$  affect simulation results in Section III.

### 2.3 Parallel Tempering

We have observed that applying multiple temperatures in a simulation is crucial for sampling the configuration space with a complex structure. Parallel tempering (PT), like simulated annealing, employs this important component. The main difference between the two methods is that whereas SA uses a fixed schedule of temperatures, PT swaps temperatures dynamically. SA is also limited to solving optimization problems. It cannot be used for sampling a distribution at a fixed positive temperature, and basic Metropolis sampling



can only do this efficiently for high enough temperatures.

PT simulates multiple replicas of a system concurrently, using a different temperature for each replica. Periodically, a pair of neighboring temperatures is selected and their configurations are swapped with a certain probability. Specifically, let the temperatures of  $M$  replicas be equal to  $T_1, T_2, \dots, T_M$ , where  $T_1 < T_2 < \dots < T_M$ . Simultaneously run  $M$  replicas of Metropolis simulation. Every  $N_{swap}$  steps, select a temperature  $T_i$ ,  $i = 1, 2, \dots, M - 1$  and exchange the configuration of  $T_i$  with that of  $T_{i+1}$  with an acceptance probability  $p$ . The probability  $p$  is related to the energy change and the difference between the reciprocal of the temperatures  $T_i$  and  $T_{i+1}$ :

$$p = \min(1, \exp(\Delta\beta\Delta E)) \quad (3)$$

where  $\Delta\beta = \frac{1}{T_i} - \frac{1}{T_{i+1}}$ ,  $\Delta E = E_i - E_{i+1}$  and  $E_i$  is the energy of the replica at temperature  $T_i$ . The swap probability  $p$  is chosen in such a way that it satisfies the detailed balance condition:

$$\begin{aligned} & P(r, \beta_i)P(s, \beta_j) \times \alpha[(r, \beta_i), (s, \beta_j) \rightarrow (s, \beta_i), (r, \beta_j)] \\ & \quad \times acc[(r, \beta_i), (s, \beta_j) \rightarrow (s, \beta_i), (r, \beta_j)] \\ & = P(r, \beta_j)P(s, \beta_i) \times \alpha[(r, \beta_j), (s, \beta_i) \rightarrow (r, \beta_i), (s, \beta_j)] \\ & \quad \times acc[(r, \beta_j), (s, \beta_i) \rightarrow (r, \beta_i), (s, \beta_j)] \end{aligned}$$

where  $\beta_i = \frac{1}{T_i}$ ,  $P(r, \beta_i)$  is the probability at state  $r$  and temperature  $T_i$ ,  $\alpha[(r, \beta_i), (s, \beta_j) \rightarrow (s, \beta_i), (r, \beta_j)]$  is the transition probability for swapping the state  $r$  and state  $s$ , and  $acc$  is the acceptance probability of the transition from state  $r$  to state  $s$ .

We describe the algorithm for PT as follows:

1. for  $j = 1, \dots, \frac{N_{total}}{N_{swap}}$
2. Run Metropolis method for all  $M$  replicas for  $N_{swap}$  steps
3. Randomly select a temperature  $T_i$  among  $T_1, \dots, T_{M-1}$ , and then perform a trial to swap the configuration at  $T_i$  with the configuration at  $T_{i+1}$ ; Sample a uniform random variable  $\xi$  on  $[0, 1]$  and accept the trial swap when  $\xi < p = \min(1, \exp(\Delta\beta\Delta E))$ .
4. return to step 1.

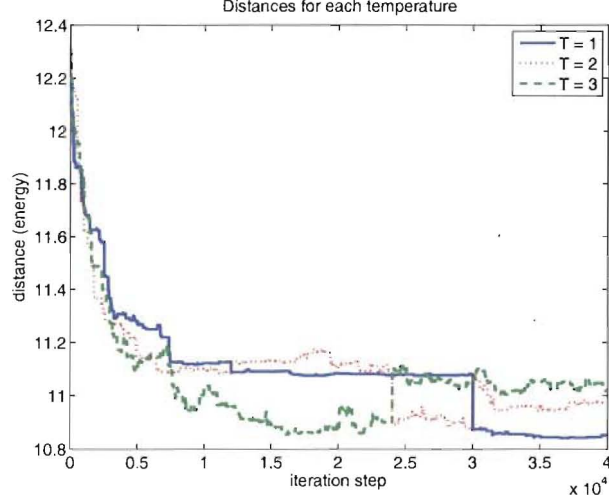


Figure 2: Tour distances at each step for temperatures  $T_1 = 0.0025, T_2 = 0.004, T_3 = 0.006$ .

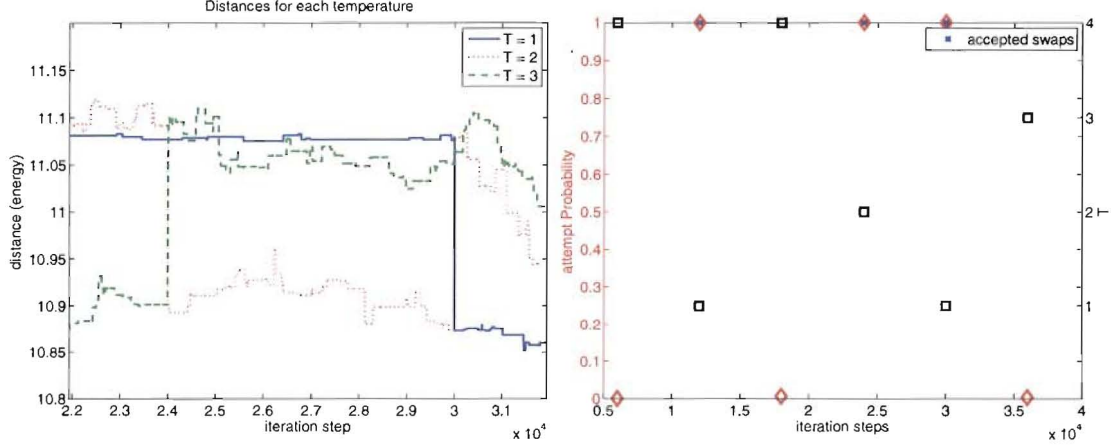


Figure 3: (Left) A blow up of Figure 2. We see jumps due to a swap of configurations between  $T_2$  and  $T_3$  at step  $\approx 2.4 \times 10^3$  and a swap of configurations between  $T_1$  and  $T_2$  at step  $\approx 3 \times 10^3$  (Right) A corresponding plot for the attempt swaps at each temperature and the attempt probabilities. Each “square” represents an attempt swap, corresponding to a “diamond” which means the attempt probability. A “cross” means a swap is accepted.

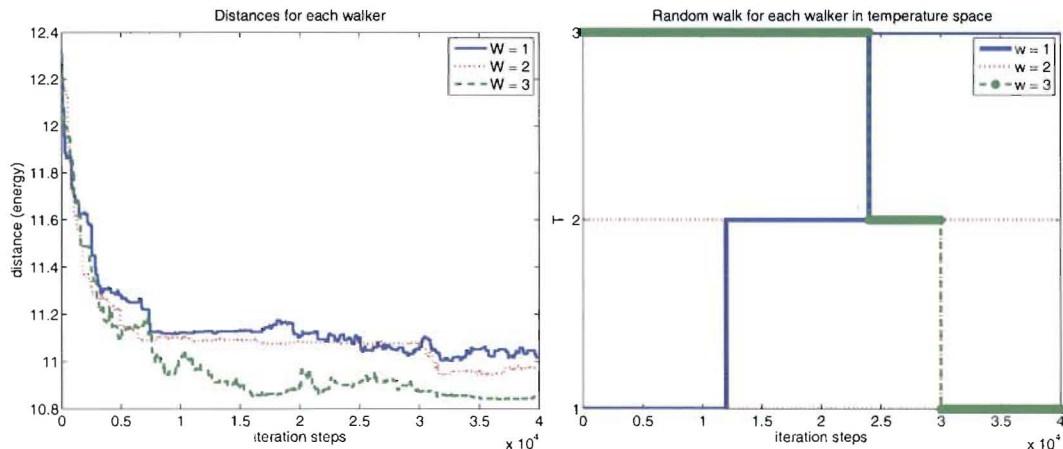


Figure 4: (Left) The tour distances at each step for fixed replicas  $W = 1, W = 2, W = 3$ . (Right) For a given replica, the swap moves create a random walk in temperature space.

There are two ways to view the temperature swaps. At any particular temperature  $T_i$ , an accepted temperature swap move creates a global update; the current state at  $T_i$  is exchanged with the state at  $T_{i+1}$ . This global change in state creates a sudden change in energy. See Figure 2 and Figure 3.

Alternatively, for a given replica the swap moves create a random walk in temperature space. When a replica drifts to a high temperature, it can overcome energy barriers and explore broad energy landscapes. When the replica returns to lower temperatures, it only moves locally in the small region. If the global minimum happens to reside in the region, small uphill moves increase the chance of finding the global minimum. Figure 4 (Left) shows the distances at each time step when the replicas are fixed. and (Right) the random walks of three replicas in temperature space. From Figure 2 to 4, we use five replicas and five temperatures. For the sake of clarity, we only show three of them here.

The values of two neighboring temperatures and their corresponding energy affect the swap acceptance rate. If  $\Delta\beta\Delta E > 0$  i.e. the replica with higher temperatures has lower energy, then the swap acceptance probability  $p$  is equal to 1. In this case, a definite temperature swap will further relax the low-energy replica; simultaneously, the high-energy replica will have a higher temperature and more likely be able to escape a local potential well.

If  $\Delta\beta\Delta E < 0$  i.e. the replica with higher temperature has higher energy,

then the acceptance probability  $p$  is less than 1.  $p$  increases as  $|\Delta\beta\Delta E|$  decreases. Such a swap gives the lower energy replica a chance to get out of the local potential well, and let higher-energy replica relax.

Our intention in implementing PT on the traveling salesman problem is to make it as analogous to SA as we can. Thus, we initialize tours for all replicas with the nearest-neighbor construction heuristic, and we use the same 2-changes for trial moves. On the other hand, PT can potentially require many more parameters to be set, including the number of replicas and the exact temperature for each replica. While there has been some study [13] of the latter, we content ourselves with employing a set of temperatures that gives satisfactory results and avoid fine tuning for the present purposes.

### 3 Simulation Results

For our simulations, we use an instance of 200 cities, distributed uniformly at random over the unit square  $[0, 1]^2$  as shown in Figure 5. We adopt periodic boundary conditions, so that the distance  $d$  between two cities at  $(x_i, y_i)$  and  $(x_j, y_j)$  is defined as

$$d_x = \min(|x_i - x_j|, 1 - |x_i - x_j|), d_y = \min(|y_i - y_j|, 1 - |y_i - y_j|)$$

$$d = \sqrt{d_x^2 + d_y^2}$$

For the instance we use, we find with the Concorde TSP solver [15] that the true optimal tour length is 10.384906. Based on this exact minimum, we calculate the “percent above minimum” for all of our SA and PT results.

In view of the uniform distribution of cities, we define the neighborhood relation in our 2-change move (for both SA and PT) as follows. A city’s neighborhood consists of all other cities that are within a distance of  $2.5/\sqrt{n}$ . Since the expected number of cities in a disc of radius  $r$  is simply  $\pi r^2 n$ , this means the neighborhood contains, in expectation,  $\pi(2.5)^2 \approx 19.635$  cities, very close to the target of 20.

Figures 6 to 8 show our benchmark results computed using SA, and Figures 9 to 10 show results computed using PT. On each Figure, 5 independent simulations, numbered 1,2,...,5, are presented.

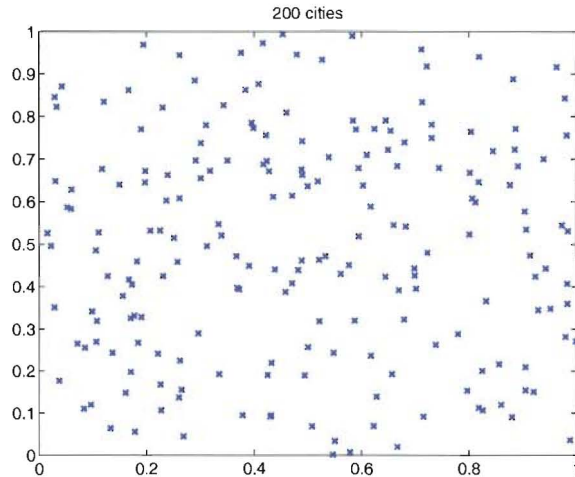


Figure 5: 200 cities uniformly distributed on  $[0, 1] \times [0, 1]$

### 3.1 Simulated Annealing

Using an implementation very similar to the one described in Section 2.2, we run SA on our  $n = 200$  instance with an initial temperature of  $T_0 = 1.5/\sqrt{200} \approx 0.1061$  and cooling rate  $r = 0.95$ , analyzing the effects of varying the total number of steps  $N_{total}$  and temperature length  $N_{length}$ . The aim is to establish a competitive baseline against which PT can subsequently be compared.

We first show that when other parameters are the same, running more steps will improve the approximation only up to a certain saturation point. Figure 6 demonstrates this effect for temperature length  $N_{length} = 10,000$ . When  $N_{total} = 10,000,000$ ,  $N_{temperatures} = \lceil \frac{N_{total}}{N_{length}} \rceil - 1 = 999$ , and

$$T_{lowest} = T_0 r^{N_{temperatures}} \approx 0.1061 \cdot (0.95)^{99} \approx 6.6123 \times 10^{-4}. \quad (4)$$

This temperature appears to be low enough that the simulation is no longer able to escape from a potential well. When we increase  $N_{total}$  to 5,000,000 and 10,000,000,  $T_{lowest}$  becomes approximately  $10^{-13}$  and  $10^{-24}$ . Such low temperatures at the end of the computation make the simulation steps essentially greedy moves. These extra steps do not improve the accuracy of the results because the simulation has already become stuck in a local minimum.

In Figure 7, we instead use a fixed computational budget  $N_{total} = 10,000,000$  but varying temperature lengths. If the simulation does not run long enough

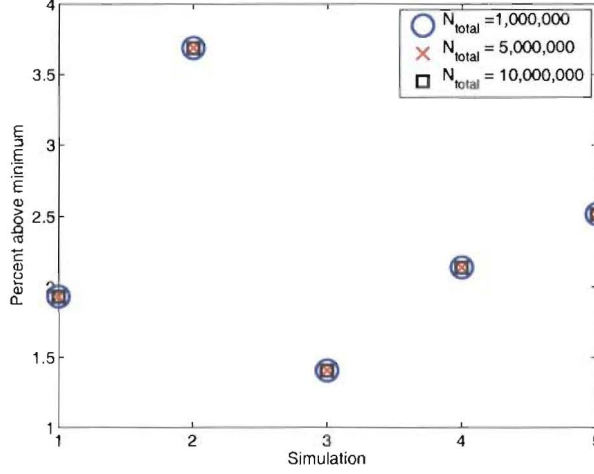


Figure 6: SA simulation: three different  $N_{total}$  values 1, 000, 000, 5, 000, 000, 10, 000, 000 are used for each of the 5 independent runs;  $N_{length} = 10,000$  for all simulations. The results computed by different  $N_{total}$  for each of the  $k$ th simulation are indistinguishable,  $k = 1, 2, \dots, 5$ .

at a given temperature, not only will  $T_{lowest}$  be unnecessarily low but the system may not even equilibrate at each temperature. Figure 7 (Left) shows that in most (but not all) cases, SA results improve as  $N_{length}$  increases, with the best results in 3 out of the 5 runs occurring for  $N_{length} = 160,000$ , where  $T_{lowest} \approx 0.0044$ . Using the temperature length parametrization of  $N_{length} = 40\alpha n$  discussed earlier, this corresponds to  $\alpha = 20$ . On the other hand, given fixed  $N_{total}$ , temperature lengths that are too long can result in  $T_{lowest}$  not being low enough. Figure 7 (Right) shows that doubling  $N_{length}$  to 320,000 ( $\alpha = 40$ ), where  $T_{lowest} \approx 0.0216$ , leads to significantly worse approximations.

From this discussion, we reach a recipe for choosing  $N_{total}$  and  $N_{length}$ . We determine the lowest temperature  $T_{lowest}$  by choosing a suitable ratio of  $N_{total}$  and  $N_{length}$ , i.e.,  $N_{temperatures}$ . The ratio must be large enough (so that  $T_{lowest}$  is small enough) to provide a good approximation to the optimal solution, but not so large as to be inefficient. Once the ratio is determined, we increase  $N_{length}$  and  $N_{total}$  simultaneously, without changing  $N_{temperatures}$ . This ensures that the simulation equilibrates at each temperature and the computation can converge to a near-optimal solution. Figure 8 demonstrates the results computed using this recipe. For each set of compu-



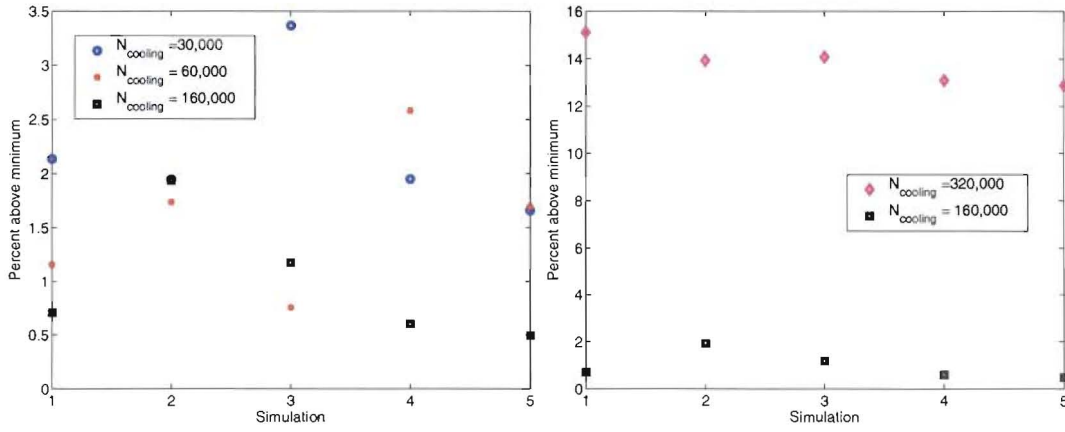


Figure 7: Five independent SA simulations using a fixed computational budget  $N_{total} = 10,000,000$  and different temperature lengths. (Left) Three different values  $N_{length} = 30,000$ ,  $60,000$  and  $160,000$ . Best results are obtained when  $N_{length} = 160,000$  is used, suggesting that simulations with smaller temperature lengths fail to equilibrate. (Right) Two different values  $N_{length} = 160,000$  and  $320,000$ . Here, the simulation with the smaller temperature length gives better results, suggesting that the larger one fails to reach low enough temperatures.

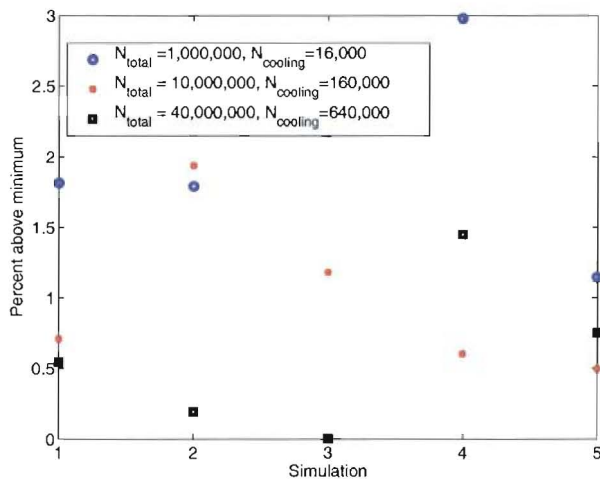


Figure 8: SA simulation: Three sets of five independent simulations for fixed  $N_{temperatures} = 62$  and corresponding lowest temperature  $T_{lowest} \approx 0.0044$ . Three different values  $N_{length} = 16,000, 160,000$  and  $640,000$  are used in the simulations, with  $N_{total} = 1,000,000, 10,000,000$  and  $40,000,000$ , respectively.

tations,  $N_{temperatures}$  is fixed at 62. When increasing the temperature length from 16,000 to 160,000 and then to 640,000, we obtain more accurate approximations to the optimal solution, in most cases well within 1% of optimal (in fact, one out of the five test simulations even hits the exact optimum). This is consistent with the results in [12].

### 3.2 Parallel Tempering

Having established benchmark results with SA, now we run our parallel tempering implementation, quantitatively demonstrating its properties by varying the input parameters. There are three sets of parameters in PT: the total number of steps  $N_{total}$ , the number of steps  $N_{swap}$  between trial swaps, and the set of temperatures  $\mathcal{T} = \{T_1, T_2, \dots, T_M\}$ . We aim to choose  $N_{swap}$  large enough so that a replica equilibrates after  $(M - 1)N_{swap}$  steps, which is the expected number of steps between when one of the  $M - 1$  neighboring replica pairs attempts a swap. For the temperatures,  $T_{max} = T_M$  should be large enough that the replica at temperature  $T_{max}$  can cross over substantial energy barriers and  $T_{min} = T_1$  should be small enough that the replica at



temperature  $T_{min}$  will approach the energy minimum. Finally, in order for PT to be effective, the spacing between temperatures must be small enough that a significant fraction of attempted swaps are accepted.

To see the effect of the choice of temperatures, let  $T_5^{low}$  and  $T_5^{high}$  be two sets of five temperatures:

$$T_5^{low} = 0.0025, 0.004, 0.006, 0.008, 0.01$$

$$T_5^{high} = 0.012, 0.014, 0.016, 0.018, 0.02$$

$T_5^{high}$  are higher than  $T_5^{low}$ . Let  $T_{10}$  be the union of the  $T_5^{low}$  and  $T_5^{high}$ ,

$$T_{10} = T_5^{low} \cup T_5^{high},$$

so that  $T_{10}$  includes a broader range of temperatures.

We find that PT clearly benefits from *both* the high and low temperatures in  $T_{10}$ . Figure 9 uses  $N_{total} = 2,000,000$  total steps and  $N_{swap} = 6,000$  steps between trial swaps. In the simulations (Left) at  $T_5^{low}$  and  $T_{10}$ , we see significant improvement when the higher set of temperatures are included in the simulation. In the context of thermal dynamics, this means the system needs the highest temperatures to avoid being trapped in potential wells and to allow exploration of broad energy landscapes. Similarly, in the simulations (Right) at  $T_5^{high}$  and  $T_{10}$ , we again see that  $T_{10}$  yields better results. Including the lowest temperatures is thus needed once a replica has entered in the basin that contains a good minimum, in order to guide it downhill toward that minimum.

We also find that unlike simulated annealing, PT can in many cases improve its simulation results simply by running more steps. The rate of improvement depends on the selection of temperatures. Figure 10 shows three simulations using the low temperatures  $T_5^{low}$  with  $N_{total} = 2,000,000$ , 20,000,000 and 40,000,000, and two simulations using  $T_{10}$  with  $N_{total} = 2,000,000$  and 20,000,000. From the three sets of results computed using  $T_5^{low}$ , we see that, although running more steps yields smaller values, the speed of the improvement is quite slow. On the other hand, if we include high temperatures and simulate using  $T_{10}$ , the approximations improve significantly. We see considerably decreased values when we run  $N_{total} = 20,000,000$  steps using  $T_{10}$ .

Finally, we note that the temperature spacing we choose is *almost* linear, but not exactly. These spacings have been chosen to try to keep swap acceptance probabilities relatively uniform (and around 20%) across the different

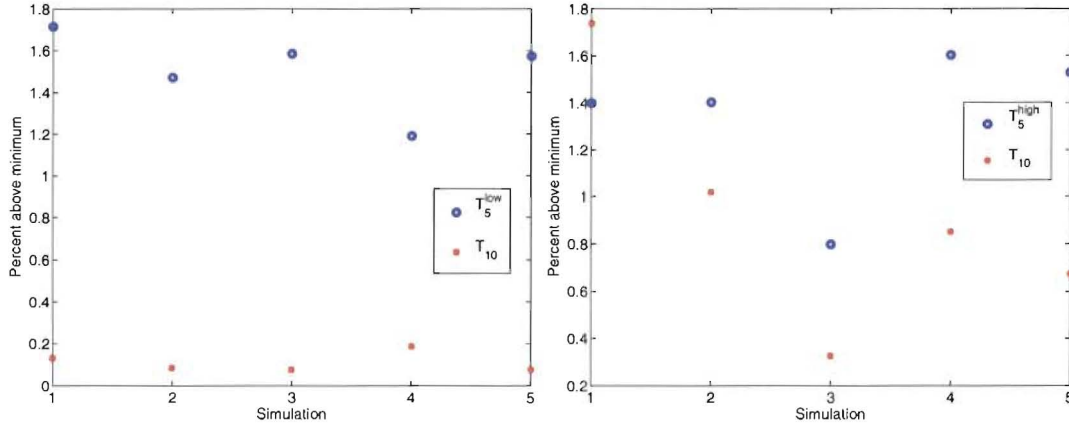


Figure 9: PT simulation: Two sets of five independent runs with  $N_{total} = 20,000,000$  and  $N_{swap} = 6,000$ , using temperatures (Left)  $T_5^{low}$  and  $T_{10}$ , and (Right)  $T_5^{high}$  and  $T_{10}$ . PT requires sufficiently high *and* low temperatures.

replica pairs. Some theory exists as to how temperature spacings should be determined [13], and this is a study in itself, beyond the scope of our work. Our aim for now is simply to show that there exists a relatively straightforward set of temperatures that allows PT to perform well.

## 4 Comparison

In the previous two sections, we have demonstrated that both SA and PT obtain good approximations of the actual minimum (well within 1%) for a sample instance of the traveling salesman problem, after suitable choice of parameters. We also find that the results computed by SA fluctuate more, while those computed by PT are more consistent. We are interested in understanding how often we can obtain such a good approximation if we repeat the simulation many times.

In particular, we compute 100 independent runs for SA and PT. Figure 11 shows the distribution (histogram) of these independent simulations of SA and PT. The parameters we use for SA are  $N_{total} = 40,000,000$ ,  $N_{length} = 640,000$ ,  $T_0 = 0.10$ ,  $r = 0.95$ , and the parameters for PT are  $N_{total} = 20,000,000$ ,  $T_{10}$ ,  $N_{swap} = 6,000$ . We confirm that the results computed by SA vary considerably more than those by PT. The SA results fluctuate between 0% and 2% above the actual optimum, and PT results fluctuate

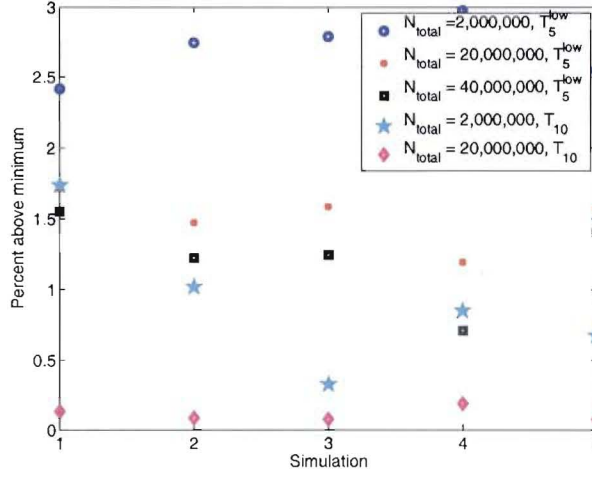


Figure 10: PT simulation: four sets of five independent runs. The first three sets use  $N_{total} = 2,000,000$ ,  $20,000,000$  and  $40,000,000$  and  $T_5^{low}$ , while the fourth set uses  $N_{total} = 20,000,000$  and  $T_{10}$ . This demonstrates that running more steps improves accuracy, but the extent of improvement may be limited by the temperature selection.

between 0% and 0.34% above the actual optimum.

In Figure 12, we examine these 100 independent runs for SA and PT in a different way from above. We group them into 20 sets of 5 runs each, and for each set choose the best of the 5. This again demonstrates that PT yields results that are more consistent than those from SA, and almost always considerably closer to optimal.

One question raised when comparing SA and PT is how to choose parameters for each method to perform an unbiased comparison. The two sets of parameters used in Figure 12 produce the best results that we were able to obtain for each method. In this comparison, PT spends five times more computational time than SA. To ensure a more unbiased comparison, we perform a simulation of SA running five times longer. Figure 13 shows the results from SA using  $N_{total} = 200,000,000$ ,  $N_{length} = 3,200,000$  (corresponding to  $\alpha = 400$ ). Interestingly, there is no noticeable improvement gained by using these larger values of  $N_{total}$  and  $N_{length}$ .

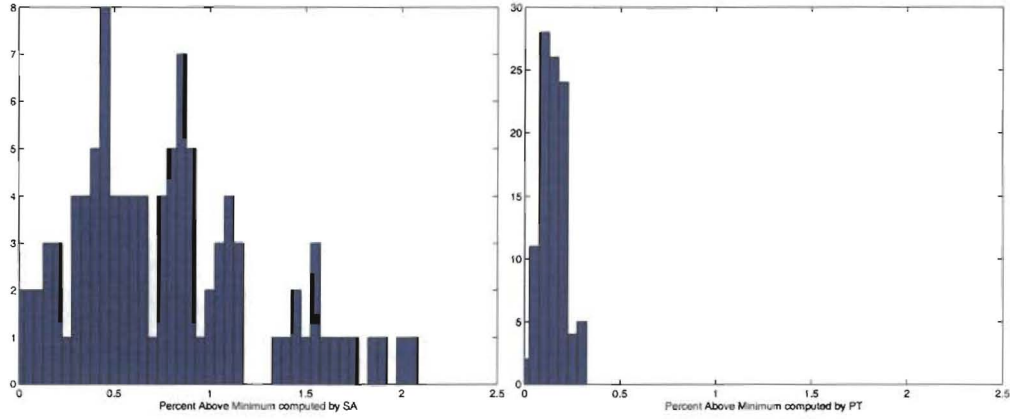


Figure 11: Histograms for SA (left) and PT (right) simulation; SA uses  $N_{total} = 40,000,000$ ,  $N_{length} = 640,000$ ,  $T_0 = 0.10$ ,  $r = 0.95$ , and PT uses  $N_{total} = 20,000,000$ ,  $\mathcal{T}_{10}$ ,  $N_{swap} = 6,000$ .

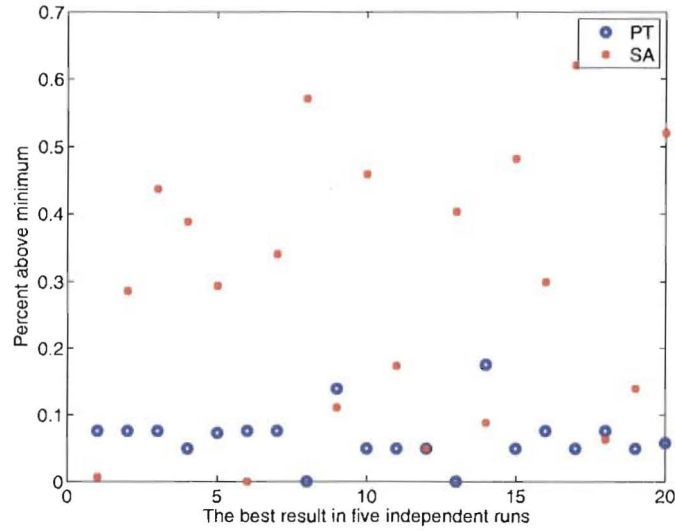


Figure 12: The best value from 5 independent runs repeated 20 times by SA and PT; SA uses  $N_{total} = 40,000,000$ ,  $N_{length} = 640,000$ ,  $T_0 = 0.10$ ,  $r = 0.95$ , and PT uses  $N_{total} = 20,000,000$ ,  $\mathcal{T}_{10}$ ,  $N_{swap} = 6,000$ .

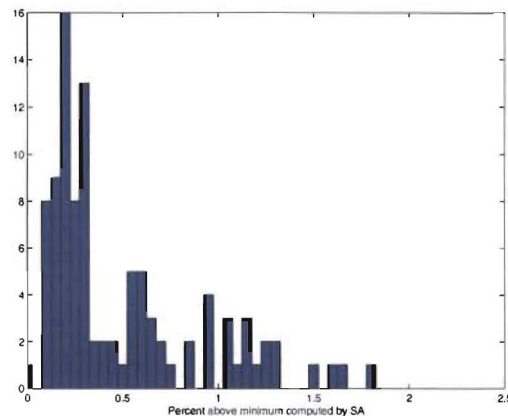


Figure 13: SA simulation with  $N_{total} = 200,000,000$ ,  $N_{length} = 3,200,000$ ,  $T_0 = 0.10$ ,  $r = 0.95$ .

## 5 Conclusion

We have presented a straightforward implementation of parallel tempering for combinatorial optimization, comparing it to benchmark results from a state-of-the-art implementation of simulated annealing. We use a traveling salesman problem instance with 200 cities distributed uniformly on a unit square, and with periodic boundary conditions. A trial move in the simulations randomly selects two cities and rearranges the original tour to obtain a new tour connecting the two cities. We find that when the parameters are chosen appropriately, both methods can closely approximate the actual minimum distance.

Moreover, our numerical study shows how the parameters for SA and PT influence the approximation, and they provide guidelines for selecting the best parameters for the two methods. For simulated annealing, we use the initial temperature and cooling rate suggested in the literature as a starting point. Our simulations show that SA requires a sufficiently large temperature length, as well as a sufficiently low (but not too low for efficiency) temperature at the end of the simulation. This means once we determine the lowest temperature, with the initial temperature and cooling rate being fixed, we should increase the total steps and temperature length simultaneously to find the optimal value. For PT, we demonstrate that the method requires sufficiently high and low temperatures to approach the optimal solution. The high temperatures are used for exploring the energy landscape, and the low

temperatures are used in finding the minimum in a local energy basin. We also show that running more time steps almost always improves PT's results. The degree of improvement depends on the temperature selection. We see minor improvement when we use the set of temperatures from 0.0025 to 0.01, but find more improvement (lower values) when using the full range of temperatures from 0.0025 to 0.02.

A significant advantage of parallel tempering is that it yields more consistent results. For example, for 100 independent simulations using the our best set of parameters, we find that the results of SA fluctuate considerably more than those of PT. This implies that it takes more simulations for SA to obtain a desired result, and that PT yields higher confidence.

A disadvantage of parallel tempering is that for the same number of steps, it takes longer to run because it concurrently simulates multiple replica. This disadvantage can be overcome by running the simulation on a parallel machine.

One caution in interpreting these results is that it is difficult to ensure that a computational comparison between the two methods is fair. Our study is biased in favor of SA because it employs knowledge from many previous experiments with SA on TSP. On the other hand, it also has a bias in favor of PT, in that more computational time was used for PT than SA. Nevertheless, we found that the SA results do not improve with additional computational time.

A major open question is how systematically to select an efficient set of temperatures for PT. The temperature spacing we use is linear, with the exception of the two lowest temperatures. From the swap acceptance probability condition, keeping the acceptance rate constant across neighboring temperature pairs means that  $\Delta\beta\Delta E$  must be kept constant. If  $E(T)$  can be replaced by its thermal average [13], which for the TSP is believed to scale as  $\sim T^2$  [16], then this suggests the gap between successive temperatures should scale as  $T^{1/2}$ . However, empirically it is not clear to us that PT actually performs as well with this prescription. We hope that our success in employing PT as an optimization algorithm will motivate further study of this question.

## References

- [1] R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett., 1986, 57, 2607.

- [2] E. Marinari and G. Parisi, Simulated tempering: a new Monte Carlo scheme, 1992 Europhys. Lett. 19 451
- [3] David J. Earl and Michael W. Deem, Parallel tempering: Theory, applications, and new perspectives, Phys. Chem. Chem. Phys., 2005, 7, 3910
- [4] G. A. Croes, A method for solving traveling salesman problems. Operations Research, 1958, 6, 791-812
- [5] M. M. Flood, The traveling-salesman problem. Operations Research, 1956, 4, 61-75
- [6] F. Bock, An algorithm for solving “travelling-salesman” and related network optimization problems. Bulletin of the Fourteenth National Meeting of the Operations Research Society of America, 1958, 897
- [7] S. Lin, Computer solutions of the traveling salesman problem. Bell Systems Technical Journal, 1965, 44, 2245-2269
- [8] S. Lin and B. W. Kernighan, An effective heuristic algorithm for the traveling-salesman problem. Operations Research, 1973, 21, 498-516
- [9] Kirkpatrick S, Gelatt C D Jr and Vecchi M P 1983 Optimization by simulated annealing, Science 220 671-7
- [10] V. Černý, Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. Journal of Optimization Theory and Applications, 1985, 45, 41-51
- [11] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth and A. H. Teller. Equation of State Calculation by Fast Computing Machines. Journal Chem. Phys. Volume 21, No. 6
- [12] D. S. Johnson and L. A. McGeoch. The traveling salesman problem: a case study, in E. Aarts and J. K. Lenstra, eds., Local Search in Combinatorial Optimization. John Wiley and Sons, London, 1997, pp. 215-310.
- [13] M. Sasaki and K. Nemoto, Application of the replica exchange method to the traveling salesman problem. Bussei Kenkyu (Study of the Condensed Matter Physics), 2000, 74, 181



- [14] D. Frenkel and B. Smit. Understanding Molecular Simulation. Academic Press.
- [15] D. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. Concorde TSP solver, available at <http://www.tsp.gatech.edu/concorde.html>
- [16] M. Mézard and G. Parisi, A replica analysis of the travelling salesman problem, Journal de Physique, 1986, 47, 1285-1296