17th Mediterranean Conference on Control & Automation
Makedonia Palace, Thessaloniki, Greece
June 24 - 26, 2009

Investigation on the choice of the initial temperature in the Simulated Annealing: A Mushy State SA for TSP

H. Shakouri G
Dep. of Industrial Engineering,
University of Tehran,
Tehran, Iran.
E-mail: hshakouri@ut.ac.ir

Kambiz Shojaee
Dept. of Electrical and Computer Engineering,
University of Tehran,
Tehran, Iran
E-mail: k.shojaee@ece.ut.ac.ir

M. Behnam T
Mechatronics Laboratory (LIM),
Politecnico di Torino,
Torino, Italy.
E-mail: mojtaba.behnam@polito.it

Abstract— **It is a long time that the Simulated Annealing (SA) procedure is introduced as a non-derivative based optimization for solving NP-hard problems. Improvements from the original algorithm in the recent decade mostly concentrate on combining its initial algorithm with some heuristic methods. This is while modifications to the method are rarely happened to the initial conditions from which the annealing schedule starts. There are several parameters in the process of annealing the adjustment of which affects the overall performance. This paper focuses on the initial temperature and proposes a lower temperature with low energy to speed up the process. Such an annealing indeed starts from a mushy state rather than a quite liquid molten material. The mushy state characteristics depends on the problem that SA is being applied to solve. In this paper the Mushy State Simulated Annealing (MSSA) is applied to the Traveling Salesman Problem (TSP). The mushy state may be obtained by some simple methods like crossover elimination. A very fast version of The Wise Experiencing Traveling Salesman is also applied to initiate SA by a low-energy-low-temperature state. This fast method results in quite accurate solutions compared to other recent novel methods.**

Keywords-component; Combinatorial Optimization, Traveling Salesman, Initial Condition

I. INTRODUCTION

Simulated Annealing (SA) is one of the earliest methods for derivative-free optimization such as Tabu Search (TS). Although it was introduced first to solve combinatorial discrete problems [1], it has recently shown a high attitude for solution of continuous problems as well [2]-[4]. SA is derived from physical behaviour of molten metals when the temperature is slowly falling to form a regular crystalline solid structure. There are two key parameters in the cooling process that determine how firm or amorphous will be the result for the metal in its frozen state. The first one is the initial temperature from which the cooling starts; and the second is the rate by which the temperature is falling.

Concerning the rate of decay, it should be low enough to allow the atoms in the molten metal to line them up and give enough time to form a crystal lattice with the minimum internal energy.

Evidently, a slow decay will lead to a long time for the solidifying process. To reduce the time, one may think of a low initial temperature. However, on the other hand, if the initial temperature is not high enough, atoms of the molten metal would not have enough freedom to rearrange their positions in a very regular minimum energy structure.

Although there are some theoretical limits and formulations to choose a proper cooling rate [5]-[14][20] , there is not any deterministic criterion to set the initial pseudo-temperature in the literature. For instance, applying SA to the travelling salesman problem (TSP), one may set it to 0.5 and change by 10% at each step [8], while some other prefer 1000 reducing by a factor of 0.99, i.e. 1% [9]. Moreover, the concept is a case dependent one and even may not fit to a bounded range, e.g. in some articles it is even initialized in a range from 0.001 to 100 [10].

There are a few research papers that suggest a formulation to relate the initial temperature to particular characteristics of the problem. Pao et al. considered an initial temperature such that the initial acceptance rate is about 70% [11]. Feng-Tse Lin, et al. proposed an Annealing-Genetic approach and use the following formula [12]:

$$T_0 = \Delta E / (\text{Population Size}/2),$$

where ΔE is the difference between the highest cost and the lowest cost found for the first generation of the randomly generated population.

Thompson and Bilbro set the initial temperature by defining a probability function for energy change in continuous problems. The probability of accepting a higher cost solution was set to 0.75. Then, the following probability distribution is solved to find T_0 [13]:

$$p = \exp(\Delta E / T),$$

where ΔE is the average cost of the random solutions plus its standard deviation.

Hao Chen et al. set the initial temperature such that the initial acceptance probability for an average uphill move is 0.97 [14].

Although SA algorithms are conceptually simple, finding optimal parameters such as initial temperature, the annealing schedule, the acceptance function parameters, etc., is by no means simple or straightforward. First of all, setting parameters for SA is problem dependent, and it is best accomplished through trial and error. Furthermore, many studies have demonstrated that SA algorithms are very sensitive to parameters, and their performances are largely dependent on fine tuning of the parameters [15]. The problem dependent nature of setting parameters for SA and its sensitivity to parameters limit the effectiveness and robustness of SA algorithms. SA possesses a formal proof of convergence to the global optima. This convergence proof relies on a very slow cooling schedule of setting the initial condition to a sufficiently large temperature and let it decay by $T_k = T_0/\log k$, where k is bound by the number of iterations[16]. While this cooling schedule is impractical, it identifies a useful trade-off where longer cooling schedules tend to lead to better quality solutions.

Also, the stochastic simulated annealing (SSA) [6] tends to find a global optimum if the annealing process is carried out sufficiently slowly. It means that SSA is able to find high-quality solutions (global optima or near-global-optima), if the temperature is reduced exponentially but with a sufficiently small exponent. For many applications, this may mean prohibitively long relaxation time in order to find solutions of acceptable quality, and conversely, reasonably long periods of time may still result in poor solutions. Lipo Wang et al. have used chaotic neural networks to be combined with the best features of SSA and have shown the effectiveness of this new stochastic chaotic simulated annealing (SCSA) [17]. However there is not any especial idea on the initializing or the cooling schedule in this approach. Before, Yuyao He had applied a chaotic noise to a Hopfield neural network and had set the annealing process such that the chaotic noise gradually reduced. Hence, it was initially chaotic but eventually convergent, and, thus, had shown richer and more flexible dynamics [18].

In brief, we observe that there is a trade off between choosing a high initial temperature or choosing a low rate of cooling, and gaining a short processing time or finding the minimum energy structure.

Exactly similar to such a trade off exists when applying SA to any optimization problem such as TSP. Assuming the objective function of an optimization problem to be an energy function, and the initial guesses for the unknown variables to be the initial problem, the above mentioned trade off appears as shown by the following notation:

Initial temperature ↑ ⇒ Optimization time ↑

Initial temperature ↓ ⇒ Final energy ↑ (Local minima)

Rate of cooling ↑ ⇒ Final energy ↑ (Local minima)

Rate of cooling ↓ ⇒ Optimization time ↑

It is easy to deduce that selection of a proper set of optimization parameters for SA itself is a multi-objective decision making (optimization) problem. In this paper we have discussed the first one, i.e. the initial temperature, and propose an approach to speed up the algorithm while obtaining accurate solutions for the chosen case study, which is TSP.

It is usual to select a very high temperature that provides a suitable initial condition with enough mobility for the atoms to move freely to new locations faraway enough in order to form as possible as minimum-energy structures. A certain criterion is to set it large enough that almost any trial point (state) will be acceptable.

This may cause the SA process to experiment new accepted points with even higher energy states. As the temperature decays, the probability to accept states that do not reduce the energy decreases.

Since the cooling process that starts from a high temperature in a liquid-like state is a time consuming, this paper proposes to start annealing from a state in a lower temperature with a lower internal energy. Such a state may be called a mushy state, rather than a liquid state. In such reduced temperatures with low energy, the ratio of acceptable states to the total trials may be less than 10%, compared to that of usual high temperatures.

After that the state is set to a lower energy state in a lower initial temperature, the annealing process can bring us the benefit of a faster local search and find the optimal state with the minimum energy. Starting from a very high temperature the metal should be cooled slowly, otherwise the atoms do not have time to orient themselves into a regular structure, but if the initial state is imposed to the atoms in a low energy low temperature, we can adjust the cooling rate to be faster.

Perhaps there are many optimization methods, even direct (random search methods) that can be applied as a prelude for SA. A simple algorithm that we are introducing in this paper is to eliminate all intersecting paths in an initially selected random tour.

II. A SHORT OVERVIEW ON THE SA

Rather than giving a detailed description of SA, herein the fundamental terminology of SA is explained shortly [7][19]. In the discrete version of SA which is used for solving TSP the set of paths traveled can be interpreted to the movement of the atoms; so the objective is the total energy or distance. Starting from an initial state in a certain temperature, T_0, the initial tour is randomly changed by a pre-defined *move set* like *inversion*, *translation*, and/or *switching*. Then the change in energy, ΔE, is calculated. For acceptance an exponential function that allows higher energy in the new state is commonly used. For example if $\exp(-\Delta E / T_k)$ is greater than a random number uniformly distributed in (0, 1), the new sequence (tour) is accepted. This acceptance of an uphill move helps the algorithm to escape from any local minima. The random change in the state is repeated several times with a certain temperature in a Markov chain. An annealing or cooling schedule regulates how rapidly the temperature T goes from high to low values, as a function of time or iteration counts.

The exact interpretation of *high* and *low* and the specification of a good annealing schedule require certain problem-specific physical insight and/or trial-and-error. The easiest way of setting an annealing schedule is to decrease the temperature T by a certain percentage at the k^{th} iteration:

$$T_{k+1} = \alpha\, T_k \qquad (1)$$

where $0 < \alpha < 1$ is an adjusting parameter. It is proved that a *Boltzmann machine* using the aforementioned generating function can find a global optimum of $f(x)$ if the temperature T is reduced not faster than $T_0 / \log(k)$ [16].

Researchers have used various cooling strategies, among which we will use the following:

$$T_{k+1}=T_k/log(k^{1/D}). \qquad (2)$$

where D is two.

III. The Traveling Salesman Problem (TSP)

The Travelling Salesman Problem (TSP) is probably the most well-known typical NP-hard problem. TSP is quite easy to describe but hard to solve. Its optimal goal is to find a closed loop of the paths with minimal total weights in a finite complete graph. Lets denote the set of "cities" in TSP as $C = \{c_1 , c_2 ,..., c_n\}$ in company with a matrix D_T an element of which is called d_{ij} that gives the distance or cost function (weight) for going from t_i, to t_j. In real problems, usually the coordinates of the cities are given, by which the matrix D_T can be easily computed.

The path linking the two cities here is called a "link". A sequence of cities $C^* = [c_{s1}, c_{s2}, \ldots, c_{sn}]$ denotes a legal solution of TSP (the salesman must visit each city once and only once), where $\{s_1, s_2, ..., s_n\}$ is a sequence of {l, ..., n}. Then the Travelling Salesman Problem's optimal goal can be expressed as minimizing the following objective function that can be interpreted as energy function:

$$E(C) = d_{s_1 s_n} + \sum_{i=1}^{n-1} d_{s_i s_{i+1}} \qquad (3)$$

where $C = [s_1, s_2, ..., s_n]$ is the travelling tour. If all the costs between any two cities are equal in both directions, i.e. D_T is a symmetric matrix, the problem is called symmetric TSP; otherwise, it is called asymmetric [21].

Sometimes D_T is calculated based on the coordinates of the cities that may generate real numbers. Normally d_{ij}'s are rounded to integer numbers to standardize the results according to the standard code proposed in [21]. Generation of the distance matrix, D_T, given the coordinates is a straight forward procedure. However, the reverse process is not possible for all cases.

Suppose the coordinates are given by two vectors namely X and Y. There are $2n$ elements in the vectors X and Y, while a symmetric D_T contains $½\, n \times (n - 1)$ elements. Each equation can be written as:

$$(x_i - x_j)^2 + (y_i - y_j)^2 = d_{ij}^2;\ i = 1, \ldots, n;\ j = i + 1, \ldots, n - 1; \qquad (4)$$

where xi , yi , xj and yj are the ith and jth elements in X and Y respectively. Therefore, solution of $n\ (n - 1)/2$ nonlinear equations available from D_T to find the $2n$ unknown coordinates in X and Y requires:

$2n \le ½\, n \times (n - 1)$,

or equivalently:

$n \ge 5$.

Thus, there will not be a unique solution for cases with more than 5 cities. However, a feasible solution will suffice to apply the above modification to all cases with $n \ge 5$. Note that the nonlinear equations should be solved just once to find the feasible solution for the coordinates X and Y. It is obvious that for an asymmetric D_T there is no solution without any extra information.

IV. The Proposed Method for Initializing SA

As mentioned, there are not many articles talking about the initial condition when the annealing process starts. The idea proposed in this paper is originated from the behavior of the metal during the annealing process. The cooling schedule is an exponential shape function of the time that can be divided into three parts. The first part is a rapidly decaying curve with a high slope in average and the last one is the ending part of the exponential function with almost frozen state. The first part should be passed as fast as possible, while complying the lower bound on the rate of cooling. And the last part has almost no significant effect on the final result. Therefore, none of these two states are of interest in this paper. The initial temperature is proposed to be selected within the middle of the curve, as shown typically in Fig. 1.

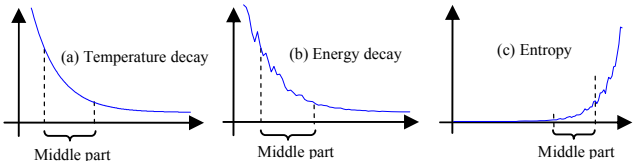


Figure 1. Typical behaviour of an annealing schedule; the mushy state falls in the middle; (a) Temperature decay, (b) Energy decay, (c) Entropy increase

To assign a proxy for the temperature, we may use a ratio named γ defined as follows:

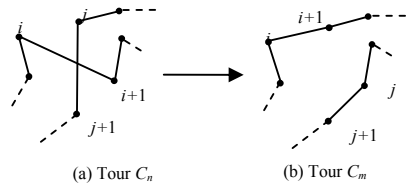
γ = the ratio of accepted new points to the total trials.

If the ratio is high, it means that the internal energy is high enough that many of motions by the atoms in a way that reduce the energy are possible. The ratio should be close to one, say 90%. Conversely, if the ratio is low, say 10%, the metal is nearby to become frozen; there are not so many new structures that reduce the energy. We propose to start annealing from such an initial condition in the middle zone, i.e. a mushy or doughty state, rather than liquid or firm states.

We have applied a simple algorithm for crossover elimination to initialize both the temperature and energy in the mushy state. The following subsections describe the Crossover Elimination method.

A. Crossover Elimination

In the special case of TSP, knowing that the optimal tour will not contain any intersection of the paths, a simple fast algorithm of intersection detection and elimination is applied. Starting from a randomly generated initial tour, every couple of links with crossover should be deleted and replace by swapping the two links. Figure 2 easily illustrates this idea.



(a) Tour C_n (b) Tour C_m

Figure 2. Intersected Links Elimination

To do so, without lacking generality, let's continue describing the algorithm for the case that the input data is given in terms of the co-ordinates. Based on this assumption, we assume that the co-ordinates are arranged in two vectors named X and Y. Now, let's assume that the initial random tour is named C_0. This vector is a sequence of the city indexes:

$$C_0 = [1, \ldots, c_i, c_{i+1}, \ldots, c_j, c_{j+1}, \ldots].$$

Suppose the subscripts of the elements of the sequence be the same shown in Fig. 2. Therefore, the line equations for all links of the tour can be calculated, by which it is possible to check that if each of the two paths are intersected or not. We need to solve $\frac{1}{2} n \times (n - 3)$ linear equations, where any valid solution should be in range of the coordinates X and Y, subsequently requiring:

$$x_{\min} < x_c < x_{\max}$$

$$y_{\min} < y_c < y_{\max}$$

where, x_c and y_c are coordinates of the intersection point of each pair of links with non-common ends, and $x_{\min}$, $x_{\max}$, $y_{\min}$ and $y_{\max}$ are the minimum and maximum coordinates found on the two links. Note that checking one of the above two conditions suffices to ensure intersection occurrence. Then, if there is a cross over, like in Fig. 1 (a), the sequence should be modified to generate T_m as follows:

$$C_m = [1, \ldots, c_i, c_j, \ldots, c_{i+1}, c_{j+1}, \ldots].$$

which is shown in Fig. 1 (b). It means that the $(i+1)^{\text{th}}$ element should be exchanged with the j^{th} element. Then it is clear that the total cost will be reduced, i.e.:

$$E(C_m) < E(C_0).$$

Once that crossover occurrence is checked $\frac{1}{2} n \times (n - 3)$ times for a tour and resolved by swapping the links, there may be new intersections generated. Therefore, the algorithm is iterated until no crossover is found in the final tour.

The algorithm is summarized in the following:

(a) Generate an initial tour randomly;

(b) Check for intersection of each path with all other non-neighbouring paths;

(c) If there is a crossover, remove it by swapping the paths;

(d) Repeat steps (b) and (c) until there is no crossover in the tour.

V. The Results

The proposed initializing method, so called Mushy State Simulated Annealing (MSSA), is applied to many benchmarks listed in TSPLIB [21]. MSSA is run 10 times with an initial condition obtained by crossover elimination. The initial conditions, i.e. the initial pseudo-temperature and the initial energy are case dependent parameters. As mentioned in Section IV, we used the ratio of accepted motions (new points), which are found based on an acceptance function, to the total number of tried motions (γ). If this ratio tends to zero, the case is in its solid state, and if it is very high (near 1), then it is in the liquid state.

Figure 3 shows relation between the pseudo-temperature and the acceptance-trial ratio for the case of eli51. It is seen that in high temperatures the ratio saturates to about 1, and in low temperatures the case has reached to its solid state with the minimum energy. Therefore, the best initial condition to start annealing is a temperature close to the melting point or the take off point in the curve, which is about the pseudo-temperature of 30 in this case.

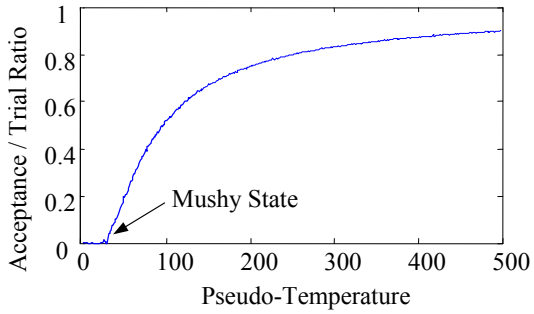


Figure 3. Relation between the Pseudo-Temperature and the Acceptance-Trial Ratio, for the case of eli51.

The results for cases with below 1002 number of cities are given below in Table 1, through which it will be easy to realize that the method has improved SA significantly. The optimal values given by the TSBLIB site, for each case are listed in the second column of the table. We have compared the best, the worst and the average of the error in the results obtained by the new approach with other results given by recent novel works (if the best and/or the worst cases are available). The error percentage is calculated by:

$$\delta = 100 (E - E^*) / E^*$$

where E^* is the optimal (minimum) energy.

The first method chosen for comparison is the Constructive Optimizing Neural Network (CONN) proposed in [22], for

which it is claimed that all runs has led to the same results, so that the best, the worst and the average of the solutions are the same. The second one is a Kohonen-Like decomposition method [23], in its three different versions abbreviated by KD, KL and KG. The third is a Genetic Algorithm-Based Clustering [24]. Finally, we have compared our results with the best and the average error percentages of the results given in [25] for its memetic neural network..

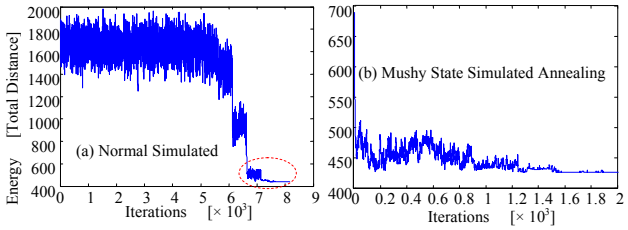


Figure 4. Energy decay in the annealing process for eli51; (a) Normal Simulated Annealing, (b) Mushy State Simulated Annealing

It is easy to deduce that MSSA by both initializing methods has led to very accurate results. To have a better feeling from the speed of the algorithms, let's first have a look at the following figures. As explained before, if annealing starts from a very high temperature, say 500 or more for the case of eli51, with more than 90% for the ratio γ (or an alternative parameter like the initial acceptance probability for an average uphill move [14]), it may take more than 8000 evaluations of the energy function to reach the minimum, while starting from a mushy state will lead to convergence in less than 2000 iterations. This means 4 times faster, as is observed in Figure 4, where it is shown how the annealing schedule would be and from which point it is started in this method. It should be added here that the total calculation time to find a mushy state with about γ =10%, done by the algorithm like crossover elimination, is less than one tenth of the total iteration time needed for SA to slow down its initially high temperature within 6000 = 8000 – 2000 iterations.

Table 1: Comparison between MSSA with other new methods for 19 benchmarks

TSP Benchmark	Optimal Solution	MSSA By Crossover Elimination			CONN			KOHONEN-LIKE Decomposition (Average δ)			Genetic Algorithm-Based Clustering (Average δ)				Memetic neural network	
		Best δ	Average δ	Worst δ	Best δ	Average δ	Worst δ	KD	KL	KG	EER	SE	ECER	SP	Best δ	Average δ
eil51	426	0	0.36	0.94	2.58	2.58	2.58	3.50	2.86	2.86	1.16	7.19	0.23	0.23	1.64	2.14
st70	675	0	1.02	1.63	2.96	2.96	2.96	3.67	1.51	2.33	-	-	-	-	0.59	0.99
eil76	538	0	0.82	1.67	5.02	5.02	5.02	6.49	4.98	5.48	4.27	3.41	0.92	2.18	2.04	2.88
gr96	514	0	0.83	1.36	3.61	3.61	3.61	-	-	-	10.62	8.07	3.09	2.46	-	-
kroA100	21282	0	0.52	0.93	2.57	2.57	2.57	-	-	-	16.54	6.70	3.81	2.41	0.24	1.14
rd100	7910	0.01	1.40	1.96	3.59	3.59	3.59	4.89	2.09	2.62	11.10	8.93	3.23	3.81	0.99	2.65
eil101	629	0	0.90	1.91	4.61	4.61	4.61	6.84	4.66	5.63	17.99	8.71	4.55	4.12	2.07	3.15
lin105	14379	0	0.52	1.00	0.38	0.38	0.38	2.18	1.98	1.29	22.55	2.85	2.31	3.10	0.00	0.34
pr107	44303	0	0.14	0.30	2.77	2.77	2.77	10.83	0.73	0.42	20.46	5.79	2.11	2.98	0.14	0.67
pr124	59030	0	0.25	0.60	1.74	1.74	1.74	3.22	0.08	0.49	30.51	3.75	2.93	3.02	0.26	1.52
bier127	118282	0.12	0.37	0.68	2.45	2.45	2.45	5.82	2.76	3.08	9.49	4.39	3.56	2.21	1.25	2.78
pr136	96772	0.35	1.05	1.97	2.27	2.27	2.27	1.93	4.53	5.15	26.50	12.54	12.43	6.19	0.73	3.10
gr137	698	0.14	0.63	1.29	4.69	4.69	4.69	-	-	-	23.85	6.58	2.54	4.22	-	-
kroA150	26524	1.36	2.18	3.37	4.78	4.78	4.78	-	-	-	29.15	7.10	8.01	5.17	1.64	2.73
kroA200	29368	0.48	1.22	2.50	4.40	4.40	4.40	5.66	5.72	6.57	40.97	10.46	7.72	5.91	1.08	2.20
lin318	42029	0.53	1.34	1.90	-	-	-	-	-	-	53.18	12.06	12.49	12.99	3.63	5.51
pcb442	50778	2.04	2.26	2.68	5.56	5.56	5.56	8.00	11.07	10.45	60.29	14.11	19.80	12.76	3.57	6.08
att532	87550	1.56	1.96	2.37	5.66	5.66	5.66	6.15	6.74	6.80	67.58	17.72	16.18	18.41	3.29	4.21
pr1002	259045	2.07	2.27	2.48	6.94	6.94	6.94	7.08	-	7.60	-	-	-	-	4.75	6.11
Average		0.46	1.06	1.66	3.70	3.70	3.70	5.45	3.82	4.34	26.25	8.26	6.23	5.42	1.64	2.84
*MSSA By Crossover Elimination		-	-	-	0.45	1.04	1.65	1.00	0.94	1.00	0.99	0.99	0.99	0.99	0.50	1.09

* Average of the cases for which the corresponding method is run and the results are given.

VI. CONCLUSION

Simulated annealing is one of the top ten methods of non-derivative based optimization methods, various versions of which are proposed by researchers during the two last decades. Focusing on the initial condition by which the annealing starts, this paper proposes a novel variant of the original SA named mushy state simulated annealing (MSSA). In this method we start annealing not from a high temperature in a liquid state, but from a low temperature in a mushy state. This technique has speeded up the optimization process while achieving to quite accurate optimum solutions. For the case study of TSP, one simple algorithm including crossover elimination is used to initiate the MSSA. Results are compared to many recent new optimization methods that are applied to solve TSP. Despite of its higher speed compared to the normal SA, superiority of the proposed method is observed in all cases with less than 1002 cities. The average error obtained by MSSA for the 19 benchmarks is less than all other methods compared to this method.

REFERENCES

[1] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", *New York: Freeman, 1979.*

[2] Herbert H. Tsang, and Kay C. Wiese "The Significance of Thermodynamic Models in the Accuracy Improvement of RNA Secondary Structure Prediction Using Permutation-based Simulated Annealing", *IEEE Congress on Evolutionary Computation (CEC 2007).*

[3] Ming-Hao Hung, Li-Sun Shu, Shinn-Jang Ho, Shiow-Fen Hwang, and Shinn-Ying Ho, "A Novel Intelligent Multiobjective Simulated Annealing Algorithm for Designing Robust PID Controllers*", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 38, NO. 2, pp. 319-330,(MARCH 2008).*

[4] Kevin I. Smith, Richard M. Everson, Jonathan E. Fieldsend , Chris Murphy, and Rashmi Misra, "Dominance-Based Multiobjective Simulated Annealing", *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 12, NO. 3, pp. 323-341, (JUNE 2008)*

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science, vol. 220, pp. 671–680, (1983).*

[6] E. Aarts and J. Korst, "Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing", *New York: Wiley, 1989*.

[7] Joshua W. Pepper, Bruce L. Golden, and Edward A. Wasil, "Solving the Traveling Salesman Problem With Annealing-Based Heuristics: A Computational Study", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 32, NO. 1, (JANUARY 2002).

[8] Hyeon-Joong Cho, Se-Young Oh and Doo-Hyun Choi, "Population-oriented simulated annealing technique based on local Temperature concept", *ELECTRONICS LETTERS Vol. 34 No. 3 pp.312-313 (5th February 1998).*

[9] Percy P. C. Yip, and Yoh-Han Pao, "Combinatorial Optimization with Use of Guided Evolutionary Simulated Annealing", *IEEE TRANSACTIONS ON NEURAL NETWORKS. VOL. 6. NO. 2, pp. 290-295 (MARCH 1995).*

[10] Andrew Soh, "Parallel N-ary Speculative Computation of Simulated Annealing", *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 6, NO. IO, pp. 997-1005 (OCTOBER 1995).*

[11] D. C. W. Pao, S. P. Lam and A. S. Fong, "Parallel implementation of simulated annealing using transaction processing", IEE Proc-Comput. Digit. Tech.. Vol. 146, No. 2, March 1999, pp. 107-113.

[12] Feng-Tse Lin, Cheng-Yan Kao, and Ching-Chi Hsu, "Applying the Genetic Approach to Simulated Annealing in Solving Some NP-Hard Problems", *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. 23. NO. 6, (NOVEMBER/DECEMBER 1993)*

[13] Dale R. Thompson and Griff L. Bilbro, "Sample-Sort Simulated Annealing", *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 35, NO. 3, pp. 625-632 (JUNE 2005).*

[14] Hao Chen, Nicholas S. Flann, and Daniel W. Watson, "Parallel Genetic Simulated Annealing: A Massively Parallel SIMD Algorithm", *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 9, NO. 2, pp.126-136 (FEBRUARY 1998).*

[15] K.L. Wong A.G.Constantinides, "Speculative parallel simulated annealing with acceptance prediction", *ELECTRONICS LETTERS Vol. 34, No. 3, 5th February 1998,* pp. 312-313.

[16] L. Ingber and B. Rosen, "Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison," *Mathematical Computer Modeling,* vol. 16, no. 11, (1992), pp. 87-100.

[17] Lipo Wang, Sa Li, Fuyu Tian, and Xiuju Fu, "A Noisy Chaotic Neural Network for Solving Combinatorial Optimization Problems: Stochastic Chaotic Simulated Annealing", *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 34, NO. 5 pp. 2119-2125, (OCTOBER 2004).*

[18] Yuyao He, "Chaotic Simulated Annealing With Decaying Chaotic Noise", IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 13, NO. 6, NOVEMBER 2002, pp. 1526-1531.

[19] J. Jang, C. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing", Proc. of the Prentice Hall 1997.

[20] G. Dueck and T. Scheuer, "Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing," *J. Computer. Phys.*, vol. 90, 1990, pp. 161–175.

[21] *G. Reinelt. Tsplib95, 1995. Available at: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95.*

[22] M. Saadatmand-Tarzjan, M. Khademi, M. R. Akbarzadeh-T., and H. Abrishami Moghaddam, "A Novel Constructive-Optimizer Neural Network for the Traveling Salesman Problem" *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 37, NO. 4, (AUGUST 2007).*

[23] Necati Aras, I. Kuban Altınel, and John Oommen, "A KOHONEN-LIKE DECOMPOSITION METHOD FOR THE EUCLIDEAN TRAVELING SALESMAN PROBLEM KNIES_DECOMPOSE", *IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 14, NO. 4,* (*JULY 2003*).

[24] Chun-Hung Cheng, Wing-Kin Lee, and Kam-Fai Wong, "A Genetic Algorithm-Based Clustering Approach for Database Partitioning*", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 32, NO. 3,* (*AUGUST 2002*).

[25] J. C. Creput, A. Koukam, "A memetic neural network for the Euclidean traveling salesman problem", *Neurocomputing Accepted 22* (*January 2008*).