Functionality limit of classical simulated annealing

To cite this article: M Hasegawa 2015 J. Phys.: Conf. Ser. 640 012001

View the article online for updates and enhancements.
Functionality limit of classical simulated annealing

M Hasegawa
Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba 305-8573, Japan
E-mail: hasegawa@esys.tsukuba.ac.jp

Abstract. By analyzing the system dynamics in the landscape paradigm, optimization function of classical simulated annealing is reviewed on the random traveling salesman problems. The properly functioning region of the algorithm is experimentally determined in the size-time plane and the influence of its boundary on the scalability test is examined in the standard framework of this method. From both results, an empirical choice of temperature length is plausibly explained as a minimum requirement that the algorithm maintains its scalability within its functionality limit. The study exemplifies the applicability of computational physics analysis to the optimization algorithm research.

1. Introduction
With the rapid development of computer capabilities, there is a growing interest in the understanding of physical and other phenomena in the energy landscape paradigm [1]. Searching for the ground state of a molecular system is minimizing its energy function, and to solve this problem, one has used general-purpose optimization methods such as simulated annealing (SA) [2, 3]. In recent computer experiments for equilibrium systems, more sophisticated methods like parallel tempering [4, 5] are generally employed instead of the conventional Metropolis algorithm (MA) [6]. While these improvements are needed to meet the computational requirement for systems with very rugged landscapes (e.g. glasses), they expand the range of options for stochastic optimization [7]. In such an interaction, we find a positive relationship between computational physics methodology and optimization algorithm design. Concerning the molecular systems, the understanding of the system dynamics in the landscape is expected to be useful, for example, for the selective and efficient formation of crystals [8]. A similar understanding should be informative in the effective and efficient implementation of the local search (LS) optimization algorithms [9].

This paper revisits classical SA with particular attention to its functionality, which is assessed through the observation of the search dynamics hidden in the landscape of the cost function. In a previous study [10], the author verified the physical analogy of SA in the solution of the traveling salesman problem (TSP) [11], a commonly-used benchmark in combinatorial optimization. There, we found that the effectiveness of this method comes not from equilibrium sampling at low temperatures, though described in its original scenario, but from downward interbasin dynamics occurring in the rugged landscape before equilibrium. Subsequent studies on the same problem [12–17] showed that the optimization characteristics of SA and SA-related algorithms can be understood more properly in a unified manner by focusing on the search function of the above relaxation dynamics. Needless to say, identification of actual success factors should
be conducive to fair competition among existing solution methods and further to practical evaluation of related new approaches like quantum annealing [18].

Here, we consider a commonly recommended, but not necessarily well-founded strategy for “temperature length [11]” in the implementation of classical SA and attempt to obtain a clear picture of its validity as an extension of the series of studies [12–17]. This strategy specifically states that the number of steps allocated at each temperature should be at least proportional to the neighborhood size in the standard geometric cooling scheme. In the practical application of SA, it has been rightly recognized and accepted that the method has an unavoidable imperfection due to the finiteness of its implementation time. In addition to this defect, the second but preventable malfunction is detected in the solution of the random TSPs [13]. As discussed below, the present empirical strategy can in fact plausibly be explained as a minimum requirement that the algorithm maintains its scalability within its functionality limit.

This work is intended neither to improve the existing solution methods nor to propose a novel algorithm. The purpose is to show that a function-based analysis using the computational physics method helps to refine our knowledge of optimization. For this reason, the algorithm is implemented even under the condition unsuited for practical use. This extensive analysis tells us that the optimization study in the landscape paradigm provides fresh insight even into the present old-established method. The paper is organized as follows: In Sec.2, the method and results of functionality assessment are presented with detailed descriptions of the problem and the algorithm. Although the same assessment has already been made in the framework of generalized SA [14], the results for classical SA are shown for the first time for completeness in the present context. In Sec.3, the influence of the second defect (described in Sec.2) on the scalability test is demonstrated using the standardized SA algorithm, and the empirical strategy for temperature length is reviewed from the functionality point of view. The study is summarized and concluded in Sec.4.

2. Functionality assessment of the relaxation dynamics
To begin with, the test-bed problem, notation, and terminology are summarized. This paper examines the solution process of the TSP, the goal of which is to find the shortest tour that passes through each of the given cities once and returns to the starting city. We consider here a random Euclidean TSP, where the locations of N cities are sampled uniformly in the unit square region and the intercity distance is computed under the Euclidean metric. (The characteristic features described below were similarly observed in a qualitative sense for a random distance matrix TSP [11], the details of which are omitted here.) Let x be a tour (a feasible solution), f(x) be the cost function defined by the tour length, and \( N(x) \) be the neighborhood function. Throughout the study, we use a 2-opt neighborhood [11], which is defined as a set of tours constructed by any change of two intercity paths from the tour x. We write \( x_n \ (n = 0, 1, 2, \ldots) \) to represent the (actual) search history and \( x^* \) to represent the incumbent solution (the best-so-far solution). A locally optimal solution is a solution that does not have any improved solution in its neighborhood. A basin is defined as a set of solutions attracted to the same locally optimal solution by a simple LS, that is, a repetition of the move to the best solution in the neighborhood. Let \( y(x) \) be the locally optimal solution inside the basin including the solution x and we refer to its cost \( f(y(x)) \) as the cost of the basin. (Obviously, \( y(x) \) can be found by a simple LS starting from the solution x; this identification method is analogous to the mapping-onto-minima approach used in studies on liquid and glass [19–21].) We denote the temperature by T; however, we often use the logarithmic temperature \( \Theta(= \log_{10} T) \).

In the assessment experiment, a simple geometric cooling scheme was adopted: The logarithmic temperature schedule \( \Theta(t) \) is described as \( \Theta_s + (t/t_e)(\Theta_s - \Theta_e) \), where \( \Theta_s \) is the initial logarithmic temperature, \( \Theta_e \) is the final logarithmic temperature, and \( t_e \) is the total search time. In what follows, the total number of search steps, \( n_e \), is often referred to instead of
solution. Plotted are the results averaged over the nearest integer. Points are the results at some representative time steps and the dashed line depicted in (b) represents the performance of a simple local search starting from a random solution. Plotted are the results averaged over $I$ search processes.

$t_e$. The values of $\Theta_s$ and $\Theta_e$ were always fixed at 1 and $-4$, respectively. These are sufficiently high and low temperatures and the cooling range includes the characteristic temperatures found previously [10]. The SA algorithm for this experiment is described as follows:

[SA1] Determine the logarithmic temperature schedule $\Theta(t)$; that is, select the value of $t_e$ ($= n_e$).

[SA2] Set $n := 0$ and generate an initial solution $x_0$ randomly. Set $\Theta := \Theta(0)$ and $x^* := x_0$.

[SA3] Implement a single step of the MA and increment $n$ by one.

[SA4] If the termination condition, $n = n_e$, is satisfied, output $x^*$ and stop. Otherwise, set $\Theta := \Theta(n)$ and return to [SA1].

In a single step of the MA in [SA3], the logarithmic temperature $\Theta$ is converted to the temperature $T$ and the following [MA1] is implemented:

[MA1] Choose a trial solution $x'_n \in \mathcal{N}(x_n)$ randomly and set $\Delta := f(x'_n) - f(x_n)$. If $\Delta < 0$, accept the trial solution and set $x_{n+1} := x'_n$; furthermore, if $f(x_{n+1}) < f(x^*)$, renew the incumbent solution, $x^* := x_{n+1}$. If $\Delta \geq 0$, accept the trial solution with the probability $\exp(-\Delta/T)$ and set $x_{n+1} := x'_n$; with the complementary probability, reject the trial solution and set $x_{n+1} := x_n$.

First, we look at the relevant features of the relaxation dynamics. In Fig. 1, the actual search dynamics described by $f(x_n)$ ($n = 0, 1, 2, \ldots$) and the hidden search dynamics described by $f(y(x_n))$ ($n = 0, 1, 2, \ldots$) are shown for various time scales for $N = 316$ ($\approx 10^{2.5}$); the results averaged over $I(=2^5)$ independent runs are plotted against the logarithmic temperature. Though not fully shown in the figure, the actual dynamics [Fig.1(a)] start from a higher level ($f(x_0) \approx 159$ at $\Theta = 1$) and the cost decreases monotonically with cooling. As $n_e$ decreases, this mechanism for optimization becomes less sufficient and as found in the hidden dynamics [Fig.1(b)], the overall change in the basin’s cost is reversed at some time scale. The monotonic decrease of $f(x_n)$ means that the incumbent solution is renewed throughout the optimization process. Consequently, the optimization performance evaluated by $f(y(x^*))$ deteriorates with decreasing $n_e$ and below the critical point, SA is defeated by a simple LS, the performance of which is represented here by $f(y(x_0))$. This is the occurrence of the second malfunction touched

Figure 1. Actual [(a) $f(x_n)$] and hidden [(b) $f(y(x_n))$] search dynamics ($N = 316$, $I = 2^5$). The abscissa represents the logarithmic temperature $\Theta$ and $R(z)$ denotes that $z$ is rounded to the nearest integer.
upon in the introduction. The amount of the deterioration is maximized at some lower time scale, below which the performance recovers to the level of the simple LS.

To describe the functioning condition of the algorithm precisely, we define its functionality by the overall displacement in the hidden search dynamics. That is, if the cost of the basin including the final solution, \( f(y(x_{n_e})) \), is smaller than that including the initial random solution \( f(y(x_0)) \), we regard the relaxation dynamics as functioning properly, and otherwise, not functioning properly. Although this definition is too crude for the performance evaluation of modern optimizers, it is appropriate here because, as seen below, elucidating the minimum requirements is essential. Repeating the above experiment for different system sizes, we assess the functionality of the algorithm for various combinations of the values of the system size \( N \) and the total number of search steps, \( n_e \). In this way, the functioning region is determined in the \( N-n_e \) plane.

This analysis was done for the other four sizes, \( N = 32(\approx 10^{1.5}) \), 100, 1000, and 3162 (\( \approx 10^{3.5} \)), in addition to the above \( N = 316 \), and all the results are shown in Fig. 2. The assessment was based on the average result over \( I \) independent runs; the value of \( I \) was selected from the range from \( 2^3 \) to \( 2^7 \) so that the amount of computation is kept within a reasonable range. Note that the number of cost evaluations instead of the Monte Carlo step is used as a unit of time. This is because the former is the usual choice in optimization studies [11] and relevant to the following discussion. We see from this figure that irrespective of the system size, there is a critical time at which the functionality definitely changes. The functioning region appears to be bounded by a straight line with a slope approximately equal to two in double logarithmic scale. Obviously, the algorithm should be implemented within the functioning region for the utilization of its intrinsic optimization function; however, this can be recognized as an unreasonable demand for the solution of larger-size instances on a given time scale. Also, it can readily be expected that the emerging limitation due to the second defect restricts the scalable implementation of the algorithm. To observe the influence of this limitation, a scalability test was actually performed on the same problem instances as in the above assessment experiment. As discussed in the next section, the test results provide a possible explanation of the empirical strategy for the temperature length.

### 3. Influence on the scalability test

The test was performed using the standardized algorithm with stepwise cooling. This was introduced for the baseline implementation in Ref.[11] and is described as follows:

**[SA21]** Select an initial temperature \( T_s \), a cooling ratio \( \gamma \), and a temperature length \( L \).

**[SA22]** Set \( n := 0 \) and generate an initial solution \( x_0 \) randomly. Set \( T := T_s \) and \( x^* := x_0 \).

**[SA23]** Implement a single step of the MA, namely, \([\text{MA1}]\) and increment \( n \) by one.
Figure 3. Total number of search steps, \( n_e \), spent in the scalability test. Plotted are the results averaged over \( I \) search processes, where the value of \( I \) is \( 2^7 \) (for \( N = 25 \) and 40), \( 2^6 \) (63, 100, and 158), \( 2^5 \) (251 and 398), and \( 2^4 \) (631, 1000, and 1585). A solid symbol is used for the case that the relaxation dynamics function properly, and an open symbol is used for the case that the second malfunction occurs. Each line is obtained for each individual test by a least-squares fit (in double logarithmic scale) of the data points represented by solid symbols.

[SA24] If the stopping criterion for the MA search, \( n = L \), is satisfied, go to [SA25]. Otherwise, return to [SA3].

[SA25] If the termination condition of the algorithm is satisfied, output \( x^* \) and stop. Otherwise, set \( T := R T \), \( x_0 := x_n \), and \( n := 0 \); and return to [SA3].

The termination condition in [SA25] is that five consecutive reductions of temperature proceed without any renewal of the incumbent solution and without the acceptance ratio going above 2%. Here, the acceptance ratio is the ratio of the case that a trial solution is accepted as the next solution in the MA search.

For the sake of discussion, we consider two forms of \( L \): One is proportional to the system size (\( L = \alpha N \)) and the other is to the square of the system size (\( L = \beta N^2 \)). After fixing the proportionality constant, the above algorithm was implemented for various system sizes; the value of \( N \) was selected from the range from 25 (\( \approx 10^{1.4} \)) to 1585 (\( \approx 10^{3.2} \)). The value of \( T_s \) was fixed at 10, which is equal to that used in the first experiment (described in Sec.2), and the value of \( \gamma \) was taken to be 0.95.

The test results are summarized in Fig. 3, where the total number of search steps (specifically the accumulated number of search steps throughout the cooling process), \( n_e \), is plotted against the system size. The results were averaged over \( I \) independent runs; the value of \( I \) was selected from the range from \( 2^4 \) to \( 2^7 \) by reference to the first experiment. The functionality was assessed in the same manner except that \( f(y(x^*)) \) was used instead of \( f(y(x_{n_0})) \). The figure shows that in the case of \( L = \beta N^2 \), all the points are almost in a line and the upward trend is uniform for both \( \beta = 1 \) and 3. The average final temperature was always higher than \( 10^{-4} \) (\( \Theta_s = -4 \)). This, taken together with the average values of \( n_e \), means the cooling rate in the average run always meet the properly functioning condition of the algorithm. Each test could be conducted appropriately beyond the selected maximum size because the slope of the depicted line is almost equal to that of the estimated boundary line in Fig.2. In the case of \( L = \alpha N \), by contrast, a change in the upward trend draws our attention in the cases \( \alpha \leq 10 \). This change appears to occur with the onset of the second defect and once it occurs, \( n_e \) increases more rapidly. In this runaway phase, it is considered that whereas the actual search dynamics do not descend to the expected cost level, the system is intensively cooled according to the temperature schedule without ever satisfying the termination condition. The intrinsic optimization function is lost in this situation and such a test makes no sense. It is highly predictable that even in the case \( \alpha = 100 \), a similar change occurs at a larger value of \( N \).

As mentioned in the introduction, the length \( L \) is recommended to be at least proportional to the neighborhood size. If we follow this strategy, only the form \( \beta N^2 \) is acceptable in the present test because the neighborhood size is \( O(N^2) \). This setting, however, has been explained...
in the literature merely as a need for worthwhile tour quality or for sufficient exploration in the neighborhood. As seen here, these explanations can be refined, at least for the present TSP instances, from the functionality point of view. That is, the form $\beta N^2$ is required for the algorithm not to be implemented outside its properly functioning region, which is a prerequisite for its scalable implementation maintaining the quality of the outcome.

As noted previously [13], the second malfunction occurs when the temperature is rapidly decreased over the effective intermediate range; therefore, it will occur even when one adopts some elaborated cooling schedule. It is uncertain whether the same defect occurs in the solution of other problems (including other TSP instances) because the functionality of the relaxation dynamics could be affected by the landscape topography. Despite the lack of generality, however, it is worth pointing out that the present function-based analysis in the landscape paradigm can complement the conventional procedure-based algorithm study. The present justification for the parameter setting of the algorithm relies on the determination of the functionality limit, which was recognized neither by the observation of the actual search dynamics depicted in Fig. 1(a) nor by the usual performance evaluation, for example, by $f(y(x^*))$ alone.

4. Conclusion
The functionality of the relaxation dynamics in optimization by classical SA was assessed for the solution of the random TSPs. The assessment was performed through the observation of the hidden search dynamics analyzed in the rugged landscape of the cost function. The properly functioning condition of the method was experimentally clarified and the scalability in its standard implementation was tested on the same problem instances. The results refined our knowledge of SA: The recommended choice of the temperature length is explained as a minimum requirement that the algorithm maintains its scalability under its properly functioning condition. The study draws our attention to an illuminating aspect of computational physics analysis in the optimization algorithm research.

Acknowledgments
This work was supported in part by JSPS KAKENHI (Grant Nos.20656017 and 24656071).

References
1. Wales D J 2003 Energy Landscapes (Cambridge: Cambridge University Press)
2. Kirkpatrick S et al 1983 Science 220 671
3. Černý V 1985 J. Optim. Theory Appl. 45 41
4. Hukushima K and Nemoto K 1996 J. Phys. Soc. Jpn. 65 1604
5. Coluzzi B and Parisi G 1998 J. Phys. A 31 4349
6. Metropolis N et al 1953 J. Chem. Phys. 21 1087
7. Schneider J J and Kirkpatrick S 2006 Stochastic Optimization (Berlin: Springer)
8. Hoffmann K H and Schön J C 2013 Eur. Phys. J. B 86 220
9. Aarts E H L and Lenstra J K (ed) 1997 Local Search in Combinatorial Optimization (Chichester: Wiley)
10. Hasegawa M 2011 Phys. Rev. E 83 036708
11. Johnson D S and McGeoch L A 1997 Local Search in Combinatorial Optimization ed E H L Aarts and J K Lenstra (Chichester: Wiley) p 215
12. Hasegawa M 2011 Comput. Phys. Commun. 182 229
13. Hasegawa M 2012 Phys. Rev. E 85 056704
14. Hasegawa M 2012 J. Phys. Soc. Jpn. 81 085001
15. Hasegawa M 2013 AIP Conf. Proc. 1518 733
16. Hasegawa M and Hiramatsu K 2013 Physica A 392 4491
17. Hasegawa M and Kim C 2014 JPS Conf. Proc. 1 019009
18. Finnila A B et al 1994 Chem. Phys. Lett. 219 343
19. Stillinger F H and Weber T A 1982 Phys. Rev. A 25 978
20. Stillinger F H and Weber T A 1983 Phys. Rev. A 28 2408
21. Stillinger F H and Weber T A 1984 Science 225 983