Applied Mathematics and Computation 217 (2011) 5754–5767

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Investigation of temperature parallel simulated annealing for optimizing continuous functions with application to hyperspectral tomography

Weiwei Cai, David J. Ewing, Lin Ma *

Department of Mechanical Engineering, Clemson University, Clemson, SC 29634-0921, USA

ARTICLE INFO

Keywords:
Global optimization
Temperature parallel simulated annealing
Exchange frequency
Critical temperature

ABSTRACT

The simulated annealing (SA) algorithm is a well-established optimization technique which has found applications in many research areas. However, the SA algorithm is limited in its application due to the high computational cost and the difficulties in determining the annealing schedule. This paper demonstrates that the temperature parallel simulated annealing (TPSA) algorithm, a parallel implementation of the SA algorithm, shows great promise to overcome these limitations when applied to continuous functions. The TPSA algorithm greatly reduces the computational time due to its parallel nature, and avoids the determination of the annealing schedule by fixing the temperatures during the annealing process. The main contributions of this paper are threefold. First, this paper explains a simple and effective way to determine the temperatures by applying the concept of critical temperature (T_C). Second, this paper presents systematic tests of the TPSA algorithm on various continuous functions, demonstrating comparable performance as well-established sequential SA algorithms. Third, this paper demonstrates the application of the TPSA algorithm on a difficult practical inverse problem, namely the hyperspectral tomography problem. The results and conclusions presented in this work provide are expected to be useful for the further development and expanded applications of the TPSA algorithm.

1. Introduction

The simulated annealing (SA) algorithm was first introduced in 1983 for solving combinatorial optimization problems [1]. Since then, it has been extensively studied, with successful demonstrations of its use for both discrete [2,3] and continuous optimization problems [4–10]. These past research efforts have shown several critical advantages of the SA algorithm over other optimization techniques. For example, it can optimize complicated problems with a large number of variables and numerous confusing local minima. In addition, the SA algorithm is insensitive to the initial guess, which is especially important when no *a priori* information about the solutions is available.

On the other hand, the disadvantages of the SA algorithm are also well-recognized. One of the primary disadvantages of the SA algorithm is its high computational cost [6,10]. Many research efforts that have focused on developing variants of the SA algorithm to reduce the computational cost [11–13] can be divided into two categories. Efforts in the first category attempt to optimize the annealing schedule [2,14–17]. However, the optimal annealing schedule is usually problem-dependent [14,15], therefore limiting the applicability of the results from these efforts. The second category involves the parallelization of the SA algorithm [12,18–21]. However, most of these parallelization schemes do not guarantee convergence. Some

* Corresponding author. Address: Department of Mechanical Engineering, Room 233, Fluor Daniel Building, Clemson University, Clemson, SC 29634-0921, USA.

E-mail address: LinMa@clemson.edu (L. Ma).

0096-3003/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2010.12.054

of the parallelization schemes that do guarantee convergence, (e.g., the speculatively parallelized SA (SPSA) algorithm [20,21]) can only achieve a maximum speedup efficiency of $\log_2(N_{proc})$, where N_{proc} is the number of processes used to implement the parallel algorithm.

The above considerations motivate the study of the temperature parallel simulated annealing (TPSA) algorithm which combines the well-established parallel tempering (or replica exchange) method [22,23] and the SA algorithm [24]. The TPSA algorithm is another parallel SA algorithm that has theoretically been proven to be convergent [12,19,25], while being able to achieve linear speedup. In addition, optimization processes occur at constant temperatures in the TPSA algorithm; therefore, the TPSA algorithm does not require an annealing schedule. Once the starting and ending temperatures (T_0 and T_N) are determined, the remaining temperatures can easily be obtained. However, the TPSA algorithm has only been studied primarily on discrete functions in previous efforts [2,19,20,24,26]. Therefore, it is the goal of this current work to conduct a systematic study of the TPSA algorithm on continuous functions. This paper first explains a simple and effective way to determine T_0 and T_N by applying the concept of critical temperature (T_C) which has been successfully demonstrated on various complicated functions in [5]. Then systematic tests of the TPSA algorithm on various continuous functions are reported, demonstrating comparable performance as well-established sequential SA algorithms.

The above studies are directly motivated by a practical application, in which a so-called hyperspectral tomography problem is desired to be solved efficiently to obtain *in situ* measurements of the temperature and concentration of chemical species [8,9]. Therefore, the application of the TPSA algorithm developed in this paper was also applied to solve the hyperspectral problem, illustrating its usefulness and potential for practical applications.

The remainder of this paper is organized as follows. Section 2 provides a detailed introduction to the TPSA algorithm. Section 3 discusses the determination of the T_0 and T_N using the concept of T_C, while Sections 4 and 5 evaluate the performance of the TPSA algorithm in terms of accuracy and computational time. Section 6 discusses the impact of other parameters important to the TPSA algorithm, including the relationship between the speedup efficiency and the number of processes (N_{proc}), and the effects of the exchange frequency (*EF*). Section 7 describes the application of the TPSA algorithm to a practical problem, where the TPSA algorithm was applied to perform tomographic inversion of hyperspectral measurements. Finally, Section 8 summarizes the paper.

2. Temperature parallel simulated annealing

The TPSA algorithm, a parallel implementation of the SA algorithm, offers two advantages over sequential SA algorithms: (1) the determination of the annealing schedule can be avoided by fixing the temperatures as constant throughout the optimization process, and (2) a reduction in computational time can be achieved. Note that under the context of the SA algorithm, the term "*temperature*" represents a *parameter* used in the algorithm, to the differentiated from the physical temperature to be measured in the hyperspectral tomography technique later in Section 7 of the paper.

The mechanism of the TPSA algorithm has been explained elsewhere under the context of combinatorial optimization [2,19], and is illustrated in Fig. 1 and briefly summarized here. First, $N + 1$ temperatures (T_0–T_N) are generated and dispatched to $N + 1$ processes. Then, each process performs an optimization procedure using a sequential SA algorithm with the assigned temperature fixed as constant. Here, a well-established sequential SA algorithm described in [6] is used. After a pre-set number of iterations on each process, the processes with adjacent temperatures (labeled as T and T' in Fig. 1) exchange their optimal solutions as shown in Fig. 1. The exchange occurs at a probability p as defined in Fig. 1. If the solution at the higher temperature exhibits a smaller function value than that of the lower temperature, the solutions are always exchanged. Otherwise, the solutions will be exchanged with a probability less than 1. The specific value of the probability is determined by the temperature difference between T and T' (labeled as ΔT) and the difference in the functional values (labeled as Δf).

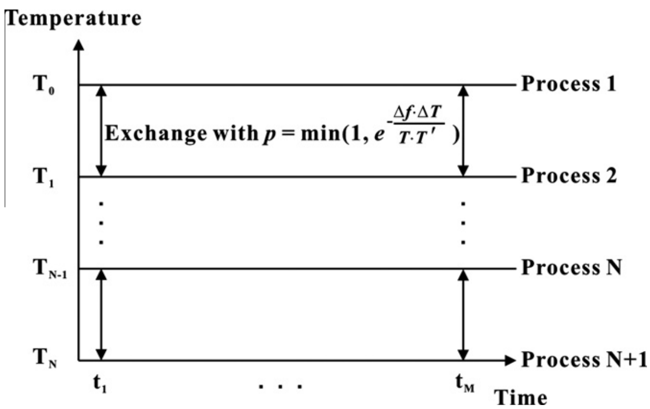

Fig. 1. Illustration of the TPSA algorithm.