Computer Physics Communications 185 (2014) 730–735

Contents lists available at ScienceDirect

Computer Physics Communications

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

Optimization by record dynamics

Daniele Barettin [a], Paolo Sibani [b,*]

[a] Department of Electronics Engineering, University of Rome, Tor Vergata, Rome, Italy
[b] FKF, University of Southern Denmark, Odense, Denmark

ARTICLE INFO

Article history:
Received 23 January 2013
Received in revised form
22 October 2013
Accepted 25 October 2013
Available online 4 November 2013

Keywords:
Annealing
Schedule
Record
Dynamics
Optimization
Algorithm

ABSTRACT

Large dynamical changes in thermalizing glassy systems are triggered by trajectories crossing record sized barriers, a behavior revealing the presence of a hierarchical structure in configuration space. The observation is here turned into a novel local search optimization algorithm dubbed record dynamics optimization, or RDO. RDO uses the Metropolis rule to accept or reject candidate solutions depending on the value of a parameter akin to the temperature and minimizes the cost function of the problem at hand through cycles where its 'temperature' is raised and subsequently decreased in order to expediently generate record high (and low) values of the cost function. Below, RDO is introduced and then tested by searching for the ground state of the Edwards–Anderson spin-glass model, in two and three spatial dimensions. A popular and highly efficient optimization algorithm, parallel tempering (PT), is applied to the same problem as a benchmark. RDO and PT turn out to produce solutions of similar quality for similar numerical effort, but RDO is simpler to program and additionally yields geometrical information on the system's configuration space which is of interest in many applications. In particular, the effectiveness of RDO strongly indicates the presence of the above mentioned hierarchically organized configuration space, with metastable regions indexed by the cost (or energy) of the transition states connecting them.

1. Introduction

Built on analogies with physical or biological processes, heuristic optimization techniques are widely used in science [1–7]. Of present interest is simulated annealing (SA), a well known local search algorithm based on the Metropolis algorithm, which minimizes the cost of candidate solutions in a way similar to a physical system minimizing its free energy under cooling [8,9]. In SA, a proposed solution is first generated by locally modifying the current solution. Changes lowering the cost are accepted and others are accepted with probability $\exp(-\Delta E/T)$, where $\Delta E > 0$ is the additional cost incurred and where the parameter T is conventionally called temperature. Ideally, a cooling schedule gradually decreasing the temperature down to zero should reach the ground state, i.e., the desired solution of the optimization problem. However, in applications to hard combinatorial problems SA invariably gets stuck in one of the many suboptimal or metastable configurations which characterize these systems. Since available local configurational changes mainly get rejected, a larger partial randomization is required to obtain further improvements.

Large changes leading a thermalizing complex system from a metastable configuration to another are often triggered by thermal energy fluctuations of record magnitude [10–13]. It is then natural to hypothesize that visiting configurations of record-high cost, or energy, similarly help a 'thermal' optimization algorithm of the SA type to escape suboptimal solutions.

The configuration space, or energy landscape, of the Edwards–Anderson spin glass [14] was previously investigated using extremal optimization [7], an optimization and exploration algorithm indifferent to energy barriers, and by the waiting time method [15], a kinetic Monte Carlo algorithm with no rejections. The analysis led to the conclusion that, in order to achieve a lower BSF value, which is desirable in optimization, the barrier $B(t)$ must previously reach a new high record. Importantly, this property is not associated with the algorithms used but pertains to all energy landscapes which can be coarse-grained into inverted binary trees where nodes represent metastable configurations [16] and height represents the energy. Motivated by the above considerations, the record dynamics optimization (RDO) algorithm introduced below dynamically generates a non-monotonic SA schedule where heating and cooling phases alternate. Each heating phase terminates once a record high 'barrier' (defined below) is encountered and each cooling phase terminates once a state of record low cost is found. Möbius et al. [17] earlier introduced a non-monotonic annealing schedule where temperature oscillations are controlled by a tunable parameter instead of being determined by intrinsic geometrical properties of the landscape.

For demonstration purposes, RDO is used to search for the ground state of a three dimensional Edwards–Anderson (EA) spin

* Corresponding author. Tel.: +45 65503525.
E-mail addresses: paolo.sibani@gmail.com, paolo.sibani@sdu.dk (P. Sibani).

0010-4655/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2013.10.030

glass [14], a standard NP hard optimization problem. For completeness, it is further applied to the two dimensional EA model. RDO performance is then compared to that of a carefully optimized version of parallel tempering (PT). The numerical effort needed to obtain results of comparable quality is similar for the two methods. However, RDO has fewer tunable parameters and is more easily implemented. Second, RDO provides, at no extra cost, some information on the configuration space structure which might be of interest in landscape explorations.

2. The RDO algorithm

First some notation: a sweep in an MC run comprises a number of elementary moves or queries, i.e. the generation and acceptance or rejection of a candidate move, equal to the number of independent variables of the problem. The number of sweeps carried out up to a certain point is dubbed time and denoted by the symbol t. Each query generates a putative solution or state, and the ordered sequence of states sampled in $[0, t]$ is called a trajectory. The cost associated with a state is called its energy E. The best so far energy, BSF(t), is the lowest energy sampled in a single trajectory in $[0, t]$. The barrier $B(t)$ associated with a state sampled at time t is $B(t) \stackrel{\text{def}}{=} E(t) - \text{BSF}(t)$. Lower case symbols are used for quantities scaled by a system size, i.e., in the example considered $b(t)$ is the barrier energy per spin. We stress that the BSF and barrier functions are stochastic processes and that inherent geometrical properties of the landscape can only be estimated by averaging over a suitably large ensemble of independent trajectories.

The RDO algorithm comprises an initial phase followed by a succession of cooling and heating phases controlled by record events. Each of these phases involves decreasing or increasing the temperature within a set of 22 predefined and equidistant values in the temperature range $[T_{\text{MIN}}, T_{\text{MAX}}]$. Several preliminary simulations showed that $T_{\text{MAX}} = 1.2$ slightly above the critical temperature of the 3d model is a good choice. Furthermore, $T_{\text{MIN}} = 0.3$ was chosen as BSF values are rarely, if at all, found below $T = 0.3$. We let the system cool and heat *ad libitum* since each cooling or heating phase produces gradually lower extremal values. Once the minimal temperature is reached, and no further BSF is found, the algorithm stops.

1. Initialization of BSF and barrier values: any short naive [9] optimization at a constant temperature typically slightly below the critical temperature T_g will produce the first BSF value, BSF_0. The first high barrier value $b_0 > \text{BSF}_0$ is found by running the algorithm at a slightly higher constant temperature. For $i = 1, 2 \ldots$, the 'barrier' $B(t) = E(t) - BSF_i$ is used to control the algorithm. The highest barrier overcome in heating phase i is called B_i.
2. Cooling: let $S_{B,i}$ be the configuration corresponding to B_i. Starting from $S_{B,i}$ run SA with decreasing temperature until a lower BSF value is found. If no lower BSF is found, cooling stops after $N_{\text{step}} = 50{,}000$ sweeps.
3. Running at constant T: the Metropolis algorithm at constant T is used until either m new BSF values have been found or the preset max time is exceeded. In practice m is a small integer, i.e., $m = 3$ in the present simulations. This step ensures that once the correct region of configuration space is identified, some time is spent exploring it. BSF_{i+1} is the lowest BSF value identified during this phase.
4. Heating: starting from S_{i+1}, the configuration corresponding to BSF_{i+1} heat the system until $B(t) = E(t) - \text{BSF}_{i+1} > B_i$. The achieved record value of $B(t)$ defines B_{i+1}.
5. Set $i + 1 \to i$, go to step 2 and repeat *ad libitum*.

3. Parallel tempering

Parallel tempering (PT) avoids trapping by independently searching a number N_T of identical replicas of the problem at hand. The m'th replica is explored by a conventional Metropolis algorithm run at a temperature T_m. Additional configurational swaps between replicas, also controlled by the Metropolis criterion in order to ensure detailed balance, provide the sought escape route from suboptimality. A successful PT implementation requires consideration of the temperatures at which the replicas are run and a compromise between the number of attempted swaps and the number of standard queries within the replicas. The reader is referred to [18–22] for an in-depth discussion of PT. The brief summary provided below describes the implementation presently used to benchmark RDO.

1. N_T different copies of the system are updated in parallel at temperatures $T_m > T_{m+1},\ m = 1, \ldots, N_T$ through one or more Monte Carlo sweeps.
2. A proposed swap between configuration C_m and C_{m+1} is accepted or rejected according to the Metropolis criterion. Defining $\beta_m = 1/T_m$, and
$$\Delta S = \left[\beta_{m+1}E(C_m) + \beta_m E(C_{m+1})\right] - \left[\beta_m E(C_m) + \beta_{m+1}E(C_{m+1})\right], \quad (1)$$
the exchange is accepted with probability $\min(1, e^{-\Delta S})$.
3. Further exchanges between the configurations associated with β_{m+1} and β_{m+2} are accepted or rejected in the same way, eventually exploring the whole set of temperatures.
4. Go to step 1 and repeat *ad libitum*.

After a number of exploratory simulations, the highest temperature was chosen as $T_{\max} = 1.6$, a value higher than the critical temperature of the Edwards–Anderson spin glass i.e. $T_c \approx 0.95$ [21]. The lowest temperature is dynamically determined as discussed below. A suitable number of temperatures for PT is generally estimated to be $N_T \approx \sqrt{N_{\text{spin}}}$ [19]. In the following, $N_T = 30$, 50 and 90 are used for $L = 30$, 50 and 100 in the 2d simulations, while $N_T = 30$, 40 and 80 are used in the 3d case for $L = 8$, 14 and 20.

To accept an exchange between copies with probability ≈ 0.5, a value considered to be optimal [20], the T_m values are treated as dynamical variables using the recursive method described in Ref. [19]. Initially, the inverse temperatures β_m are set to

$$\beta_m = \beta_1 + (\beta_M - \beta_1)\frac{m-1}{M-1} \quad (2)$$

with $M = N_T$. The updated set $\{\beta'_m\}$ is obtained using the sampled exchange rates p_m between configurations at inverse temperatures β_m and β_{m-1}:

$$\beta'_1 = \beta_1$$
$$\beta'_m = \beta'_{m-1} + (\beta_m - \beta_{m-1})\frac{p_m}{c} \quad \text{with } m = 2, \ldots, M.$$
$$c = \frac{1}{M-1}\sum_{m=2}^{M} p_m. \quad (3)$$

While in Ref. [19] temperatures are only updated initially to reach the constant values used in the simulation, we found it more convenient to update them during the simulation itself, at logarithmically equidistant times $2^n \times 100$ MC sweeps, with $n = 1, 2, \ldots, N$.

Two different benchmarks for RDO are provided. The first, our 'fast' PT, has $N = 10$ and $N_{\text{step}} = 102{,}400$ sweeps per replica. Adding the computational effort for all replicas, PT is eight time faster than RDO but produces results of somewhat lesser quality. The second version, 'slow' PT, has $N = 13$ and $N_{\text{step}} = 819{,}200$, with the total number of sweeps approximately corresponding to that used in our RDO implementation. Both versions of the PT algorithm include a final quench to $T = 0$, a step omitted in RDO. Importantly, the PT versions implemented are carefully optimized and based on the recent literature on the subject [18–22].