Optimizing Searches via Rare Events

Andrea Montanari[1,*] and Riccardo Zecchina[2,†]

[1]*Laboratoire de Physique Théorique de l'ENS, 24 rue Lhomond, 75231 Paris cedex 05, France*
[2]*Laboratoire de Physique Théorique et Modèles Statistiques, Université Paris Sud, 91405 Orsay, France*
(Received 9 December 2001; published 10 April 2002)

Randomized search algorithms for hard combinatorial problems exhibit a large variability of performances. We study the different types of rare events which occur in such out-of-equilibrium stochastic processes and we show how they cooperate in determining the final distribution of running times. As a by-product of our analysis we show how search algorithms are optimized by random restarts.

Recent years have witnessed an increasing convergence of research themes coming from out-of-equilibrium statistical physics and computer science [1–3]. For instance, giving a "static" characterization of systems displaying an extremely slow dynamics is a central problem both in computer science [4] and in spin glass theory [5]. The results in these fields are strongly focused on the *typical* properties of large random systems. This approach is justified as long as the quantities of interest concentrate in probability around some typical value when the size diverges (the so-called self-averaging property).

In this Letter we provide an analytical and numerical study of different types of "rare events" which occur in the time evolution of randomized search algorithms for hard optimization problems. As a by-product, we find a general picture for understanding and optimizing the introduction of restarts in randomized search algorithms [6].

It is a well-known fact (and a basic problem for both theoretical and applied computer science) that the so called NP-complete [7] combinatorial decision problems might require computational resources that grow exponentially with the number of variables N needed for their encoding. However, combinatorial search methods often exhibit remarkable variability in performance: it is not uncommon to observe a combinatorial method "hang" on a given instance of a problem, whereas a different heuristic algorithm, or even just another stochastic run, solves the instance quickly.

With the aim of clarifying such behavior, in recent years there has been intense research activity on randomly generated hard combinatorial problems which has lead to the identification of nontrivial problem ensembles [8]. Particularly representative and widely studied examples are satisfiability of random Boolean expressions, vertex coloring and covering of random graphs, and number partitioning [1].

This type of setting gives us much more freedom than in a standard physical experiment. Indeed an algorithm can be run an exponential number of times with each run in turn possibly taking exponential time. In such a situation rare events may have dramatic effects and completely determine the total computational time and the outcome of such random-restart experiments. We will show that there exist distinct sources of hardness fluctuations, static (i.e., intrinsic) and dynamic (algorithm dependent), which account for the variability of resolutions times.

While our approach is general and applies to a wide class of problems, in what follows we focus on the (NP complete) vertex cover (VC) problem on random graphs. The choice of VC is dictated mainly by its relative simplicity. Moreover, we will study random restart experiments for ensembles of problems which are typically solvable.

In combinatorial decision problems one has to assign the values of a (large) set of variables in such a way as to satisfy a set of constraints. For instance, in VC one is given an undirected graph $G = (V, E)$ with N vertices $i \in V = \{1, 2, \ldots, N\}$ and L edges $(i, j) \in E \subset V \times V$. The problem consists in distributing X covering marks over the vertices in such a way that every edge of the graph is covered, that is it has at least one of its ending vertices which is marked. If such covering can be found the graph is said to be coverable (COV), otherwise it is uncoverable (UNCOV).

The backtrack algorithm proceeds by choosing at random one or more variables at a time and assigning their values according to some heuristics. The problem is then turned into a subproblem in which the assigned variables act as correlated quenched randomness. For instance, in VC one declares *covered* or *uncovered* a vertex at a time, therefore reducing the size of the graph to be treated.

This evolution can be described macroscopically [9] by keeping track of a proper set of average quantities $\vec{\alpha}$ (e.g., the ratio of the number of nonsatisfied constraints to the number of nonassigned variables). The subproblem generated by fixing a fraction t of the N variables may, however, not have any solution (this happens because the algorithm made a wrong assignment at an earlier stage). Sooner or later the algorithm detects the inexistence of solutions compatible with the variables assigned so far and begins a backtrack correction process which may take an exponential number of steps to correct early mistakes.

The fraction t of assigned variables acts therefore as a control parameter and the system undergoes a SAT-UNSAT phase transition (i.e., a transition from a

satisfiable to an unsatisfiable instance of the problem) when t crosses some critical value t_c [9]. This corresponds to the trajectory $\vec{\alpha}(t)$ crossing a critical surface in the static phase diagram at $\vec{\alpha}_c \equiv \vec{\alpha}(t_c)$. For simple enough randomized algorithms it is possible to compute the size $e^{N\Omega(\vec{\alpha})}$ [9,10] of the backtracking tree for an UNSAT instance characterized by the parameters $\vec{\alpha}$. The computational complexity of the algorithm is therefore given by $\exp[(1 - t_c)\Omega(\vec{\alpha}_c)N]$.

How do rare events enter this scenario and how to take advantage of their presence? The idea is that very lucky (or unlucky) assignments of the variables in the early stages of a run (i.e., in the prebacktracking regime) can have dramatic effects on the total running time. In the case of VC one may randomly chose to cover some very important vertex in one of the first iterations, leading to a great simplification of the remaining subgraph. These rare events can take two distinct forms in the general picture described above.

(I) Let us assume that the trajectory $\vec{\alpha}(t)$ follows the most probable line. Once the SAT-UNSAT critical line is crossed there still exists a small probability $\exp[-N(1 - t)\psi(\vec{\alpha}(t))]$ of having generated a subproblem which is solvable. The deeper one goes into the UNSAT phase, the smaller will be such probability. On the other hand such a rare event corresponds to a reduction of order N of the size of the problem and, therefore, to an exponential reduction of the size $\exp[N(1 - t)\Omega(\vec{\alpha}(t))]$ of the backtracking tree.

This trade-off can be exploited in a random restart algorithm: we interrupt the search after $e^{N\tau_R}$ backtracking steps and rerun it (with different random numbers). The probability of finding a solution in one of such runs is given by $P_S \approx \exp[-N\min_t(1 - t)\psi(\vec{\alpha}(t))]$, where t is constrained by the fact that the size of the backtracking tree must be smaller than $e^{N\tau_R}$: this implies $(1 - t)\Omega(\vec{\alpha}(t)) \leq \tau_R$. Assuming that different stochastic runs quickly lead to uncorrelated subproblems, a solution is found after $N_R \approx 1/P_S$ restarts. The complexity of the algorithm is therefore $\exp[N\tau(\tau_R)]$, where

$$\tau(\tau_R) = \tau_R + \min_t(1 - t)\psi(\vec{\alpha}(t)), \qquad (1)$$

and where the minimizing value of t must satisfy

$$(1 - t)\Omega(\vec{\alpha}(t)) \leq \tau_R. \qquad (2)$$

(II) The above scenario is, however, largely incomplete. Indeed there exists another, dynamical, source of fluctuations: $O(1)$ fluctuations with respect to the typical trajectory (this effect has been studied and pointed out to us by Cocco and Monasson [11]). At time t the macroscopic parameters take the value $\vec{\alpha}$ with probability $\exp[-NI_t(\vec{\alpha})]$ [with $I_t(\vec{\alpha}) = 0$ along the typical trajectory $\vec{\alpha} = \vec{\alpha}(t)$]. Again, such a rare event implies an exponential change in the computational complexity, and the possible gain can be exploited by the random restart algorithm. Equation (1) must be properly generalized. We get

$$\tau(\tau_R) = \tau_R + \min_{t,\vec{\alpha}}\{I_t(\vec{\alpha}) + (1 - t)\psi(\vec{\alpha})\}, \qquad (3)$$

always with the constraint Eq. (2). These are not the only sources of fluctuations, but they give a quite accurate picture of the phenomenon.

Let's now apply the above scheme to the case of VC. A nontrivial ensemble of graphs which captures some relevant computational features of VC at the level of typical or average cases, is the set of random graphs $G_{N,L}$ with N vertices and L edges (and flat probability distribution). Similarly to other random NP-complete problems [1], a threshold phenomena occurs as the control parameter X is changed [12]. For a given average connectivity $c = 2L/(N - 1)$, when the number $X = xN$ of covering marks is lowered the model undergoes a COV-UNCOV transition at some critical density of covers $x_c(c)$ for $N \rightarrow \infty$. Statistical mechanics methods allow for a precise estimate of $x_c(c)$ [13] and probabilistic tools provide rigorous lower and upper bounds for such a threshold [14]. For $x > x_c(c)$, vertex covers of size Nx exist with probability one, for $x < x_c(c)$ the available covering marks are not sufficient. The statistical mechanics analysis is performed by mapping the optimization problem onto a zero temperature disordered system with Hamiltonian [12]

$$H(\{n\}) = \sum_i \delta_{n_i,0}, \qquad (4)$$

where $n_i \in \{0, 1\}$ ($n_i = 0$ if a mark is put on vertex i) and satisfy an excluded volume constraint: if $(ij) \in G$ then either $n_i = 0$ or $n_j = 0$. The ground state energy E_{GS} of the model is the minimum number of marks needed for covering the graph: for $X \geq E_{GS}$ the graph is COV, while for $X < E_{GS}$ it is UNCOV.

It is known experimentally and analytically for some algorithms [10] that the typical computational cost, given, e.g., by the number of visited decision nodes in the backtracking tree, becomes exponential for initial conditions in a region close to or below $x_c(c)$, while it remains linear well inside the coverable phase, $x > x_c(c)$. This easy-hard scenario characterizes the typical-case complexity pattern found in other NP-complete random ensembles [8].

We consider the following backtrack algorithm [15]. During the computation, a vertex can be *covered, uncovered,* or just *free,* meaning that the algorithm has not yet assigned any value to that vertex. Here is the recursive procedure: The algorithm chooses a vertex i at random among those which are *free* (at $t = 0$ all vertices are *free*). If i has neighboring vertices which are either *free* or *uncovered,* then the vertex i is declared *covered* first. In case i has only covered neighbors, the vertex is declared *uncovered*. The process continues unless the number of covered vertices exceeds Nx. If the algorithm backtracks, then the opposite choice is taken for the vertex i unless this corresponds to declaring *uncovered* a vertex that has one or more *uncovered* neighbors. The algorithm halts if it finds a solution (and declares the graph to be COV) or after exploring all the search tree (in this case it declares the graph to be UNCOV).

In order to study the prebacktracking regime we need to follow the variables X, N, L which become time dependent. In one time step ($T \to T + 1$), the probability for a change $L \to L + \Delta L$ in the number of links and $X \to X + \Delta X$ in the number of available covering marks reads

$$P_{\Delta L, \Delta X} = e^{-c} \left[\delta_{\Delta X, 0} \delta_{\Delta L, 0} + \sum_{k=1}^{\infty} \frac{c^k}{k!} \delta_{\Delta X, -1} \delta_{\Delta L, -k} \right]. \tag{5}$$

This defines a Markov process in the space $(X, G_{N,L})$ which mimics the effects of the algorithm. We want to iterate the above step ΔT times and compute the corresponding transition probability. Let us introduce the rescaled time $t = T/N$ (i.e., the fraction of assigned variables) and the macroscopic time dependent parameters $c(t) = 2L_T/N_T$ and $x(t) = X_T/N_T$ (which we denoted collectively by $\vec{\alpha}$ in the general description of our approach). Because of the Markovian structure of this process, the probability for a trajectory $\vec{\alpha}(t) \equiv [c(t), x(t)]$ can be written in a path integral form. To the leading order we get $P[c, x] = \int \mathcal{D}s \exp\{-N \int dt \mathcal{L}_t(c, x, s)\}$, where

$$\begin{aligned}
\mathcal{L}_t(c, x, s) = &-\tfrac{i}{2} s \partial_t[(1 - t)c] + (-\dot{\tilde{x}}) \log(-\dot{\tilde{x}}) \\
&+ (1 + \dot{\tilde{x}}) \log(1 + \dot{\tilde{x}}) \\
&+ \dot{\tilde{x}} \log[\exp(ce^{is}) - 1] + c,
\end{aligned} \tag{6}$$

where we used the shorthand: $\tilde{x}(t) \equiv (1 - t)x(t)$. The transition probability $P_{t_1}(c_0, x_0 \to c_1, x_1)$ is given by the corresponding constrained path integral. Such an integral can be computed by saddle point, leading to an explicit formula for the trajectories:

$$c(t) = \frac{c_0}{1 - t} - \frac{2}{B(1 - t)} \int_{e^{B(1-t)}}^{e^B} d\zeta \, \frac{\log \zeta}{\zeta - A}, \tag{7}$$

$$x(t) = \frac{x_0 - t}{1 - t} - \frac{A - 1}{AB(1 - t)} \log \left\{ \frac{1 - Ae^{-B}}{1 - Ae^{-B(1-t)}} \right\}, \tag{8}$$

where the two integration constants (A and B) must be computed from the conditions $c(t_1) = c_1$, $x(t_1) = x_1$. The large deviation functional is $I_{t_1}(c_1, x_1) = \int_0^{t_1} dt \mathcal{L}_t(\cdot)$, where the integral is computed along the trajectory (7),(8). For $A = 0$ and $B = c_0$ we recover the typical trajectory [16] and we get $I_t(c, x) = 0$. As shown in Fig. 1, numerical simulations are in remarkable agreement with the analytical predictions.

A subgraph generated according to the process described above can be still COV (with an exponentially small probability) after the trajectory $\vec{\alpha}(t) = (c(t), x(t))$ has entered the UNSAT phase [i.e., after $x(t) < x_c(c(t))$]. Repeated restarts can exploit this rare event. The size $\exp[N(1 - t)\Omega(\vec{\alpha}(t))]$ of the backtrack tree at any point in the UNCOV region can be computed analytically [10] and used in Eq. (2). Hence, in order to evaluate Eq. (3), we just need to compute the probability of being COV in the UNCOV region, that is we need to know the probability
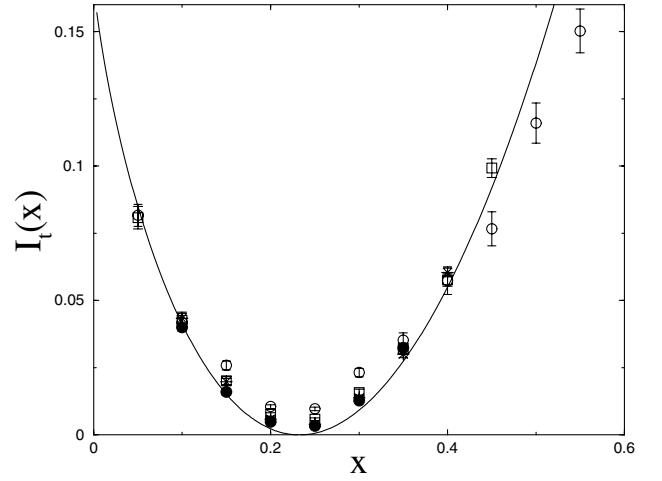

FIG. 1.   Dynamical rare events. We consider the probability $P_t(x)$ that, after Nt steps we are left with Nx marks, and we plot $I_t(x) = -\log P_t(x)/N$. The continuous line is the theoretical prediction $I_t(x) = \min_c I_t(c, x)$, while the symbols are numerical results for $N = 100$ (empty circles), 200 (squares), 300 (stars), and 400 (full circles). We used the initial condition $c_0 = 2$, $x_0 = 0.5$, and $t = 0.5$.

distribution of the ground state energy of the model (4). This computation can be carried over by the replica method. We notice that the replicated partition function averaged over the disorder reads

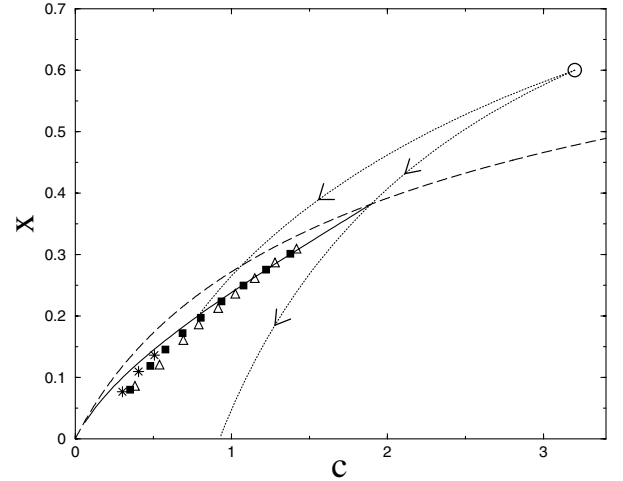$$\langle Z^n \rangle \to \int_0^{\infty} dP(E_{GS}) e^{-\omega E_{GS}}, \tag{9}$$

FIG. 2.   Random restart experiments with initial condition at $c_0 = 3.2$, $x_0 = 0.6$ (empty circle). The long dashed line is the replica symmetric critical line $x = x_c(c)$. The rightmost dotted line is the typical trajectory. The leftmost one is the rare trajectory followed by the last (successful) restart of the algorithm when $\tau_R = 0.1$. The symbols are numerical results for the (c, x) coordinates of the root of the backtrack tree generated by the algorithm since the last restart. Triangles, squares, and stars correspond, respectively, to $N = 30, 60$, and 120 (in each case we considered several values for τ_R, each one corresponding to a symbol). The continuous line is the theoretical prediction for the same quantity [i.e., the minimizing $\vec{\alpha} \equiv (c, x)$ in Eq. (3)].
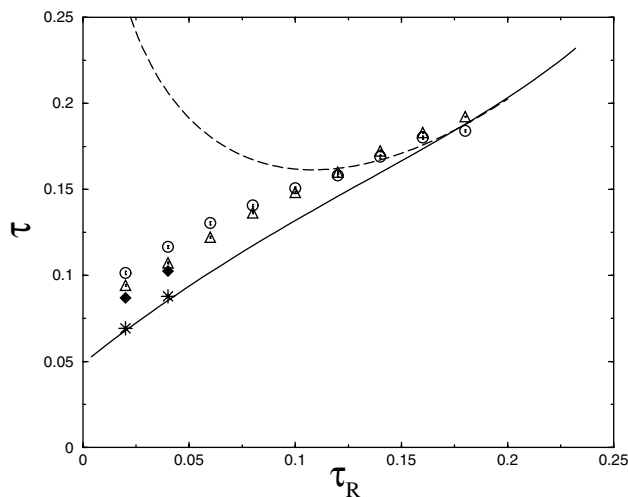
FIG. 3. Typical computational complexity of the random restart algorithm. Here we plot the logarithm of number of nodes visited by the algorithm divided by the size N, for different values of the restart rate τ_R. Symbols refer to $N = 30$ (circles), 60 (triangles), and 120 (diamonds). The stars are the result of an $N \rightarrow \infty$ extrapolation. The continuous and dashed lines reproduce the theoretical prediction with, cf. Eq. (3), and without, cf. Eq. (1), dynamical rare events.

where one takes the zero temperature limit $\beta \rightarrow \infty$ keeping $\omega \equiv n\beta$ fixed. As $N \rightarrow \infty$, $\langle Z^n \rangle \sim \exp[-N\phi(\omega)]$ and $P(E_{GS}) \sim \exp[-N\psi(x)]$, where $x = E_{GS}/N$. We get therefore $\phi(\omega) = \psi(x) + \omega x|_{\omega = -\partial_x\psi}$.

The small ω behavior of $\phi(\omega)$ yields the typical ground state energy and its small fluctuations. The knowledge of the whole function $\phi(\omega)$ gives the large deviation properties of the ground state energy. In general $\psi(x)$ is convex and has its unique zero at the typical ground state energy $x = x_c(c)$. The probability that a graph in the ensemble is coverable with $X = Nx < Nx_c(c)$ marks is given by $\exp[-N\psi(x)]$. A replica symmetric calculation yields

$$\phi(\omega) = c(1 - F_Q) + \frac{c}{2}\log F_Q$$
$$- \log[e^{-\omega} + (1 - e^{-\omega})e^{-cF_Q Q}], \quad (10)$$

where we used the short-hand $F_Q = [1 + (e^{-\omega} - 1)Q^2]^{-1}$ and Q satisfies the self-consistency equation: $Q^{-1} = 1 - e^{-\omega}[1 + \exp(cF_Q Q)]$. Figure 2 gives the geometrical picture of a random restart experiment. Quite remarkable is the prediction on the (c, x) values up to which the algorithm has to backtrack before finding the solution. Such a curve lies well inside the UNCOV region indicating that the two types of rare events are both relevant for $\tau_R > 0$.

In Fig. 3 we consider the computational complexity $e^{N\tau(\tau_R)}$ of the random restart algorithm for the initial condition $c = 3.2$, $x = 0.6$. Finite size effects are important for the achievable sizes of the problem. An extrapolation can be done for the smaller values of τ_R, where we were able to run the algorithm on much larger systems.

Building on large deviations results we have shown that running times of randomized complete search algorithms can be exponentially reduced by a restart strategy. The

optimal restart rate τ_R^{opt} can be computed within our approach: for VC we find $\tau_R^{\mathrm{opt}} = 0$. This implies restarting after a subexponential — but at least linear — running time. The computational complexity of this strategy is given by the probability $P_0 = \max_t P_t(c_0, x_0 \rightarrow 0, 0)$ (for $c_0 = 3.2$, $x_0 = 0.6$, we get $P_0 \sim e^{-0.049N}$) of trajectories which do not cross the critical line. For a thorough investigation of the subexponential regime, see [11]. In more general cases we expect $\tau_R^{\mathrm{opt}} > 0$ [6].

---

*Electronic address: montanar@lpt.ens.fr
†Permanent address: ICTP, Strada Costiera 11, I-34100 Trieste, Italy. Electronic address: zecchina@ictp.trieste.it

[1] Special issue on Phase Transitions in Combinatorial Problems, edited by O. Dubois, R. Monasson, B. Selman, and R. Zecchina [Theor. Comp. Sci. **265**, Nos. 1, 2 (2001)].
[2] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing* (Oxford University Press, Oxford, United Kingdom, 2001).
[3] M. Mézard, "Theory of Random Solid States" (unpublished).
[4] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, Nature (London) **400**, 133 (1999).
[5] J.-P. Bouchaud, L. F. Cugliandolo, J. Kurchan, and M. Mézard, in *Spin Glasses and Random Fields,* edited by A. P. Young (World Scientific, Singapore, 1997).
[6] C. P. Gomes, B. Selman, N. Crato, and K. Kautz, J. Autom. Reasoning **24**, 67 (2000); C. P. Gomes, B. Selman, and H. Kautz, in *Proceedings of AAAI-98, Madison, WI, 1998* (AAAI Press, Menlo Park, CA, 1998), pp. 431–437.
[7] M. R. Garey and D. S. Johnson, *Computers and Intractability* (Freeman, New York, 1979).
[8] Special issue on Frontiers in Problem Solving: Phase Transitions and Complexity, edited by T. Hogg, B. A. Huberman, and C. Williams [Artif. Intell. **81**, Nos. I, II (1996)].
[9] S. Cocco and R. Monasson, Phys. Rev. Lett. **86**, 1654 (2001); Eur. Phys. J. B **22**, 505 (2001).
[10] M. Weigt and A. K. Hartmann, Phys. Rev. Lett. **86**, 1658 (2001).
[11] S. Cocco and R. Monasson (to be published).
[12] M. Weigt and A. K. Hartmann, Phys. Rev. Lett. **84**, 6118 (2000); Phys. Rev. E **63**, 056127 (2001).
[13] For $c < e$, the statistical mechanics analysis [12] and algorithmic results [14] give $x_c(c) = 1 - (2W + W^2)/2c$, where $We^W = c$. For $c > e$ there appear replica symmetry breaking effects.
[14] P. G. Gazmuri, Networks **14**, 367 (1984); A. M. Frieze, Discrete Math. **81**, 171 (1990); M. Bauer and O. Golinelli, Eur. Phys. J. B **24**, 339–352 (2001).
[15] R. Motwani and P. Raghavan, *Randomized Algorithms* (Cambridge University Press, Cambridge, United Kingdom, 2000).
[16] $c(t) = c_0(1 - t)$, $\quad x(t) = \{x_0 - t + \frac{1}{c_0}[e^{-c_0(1-t)} - e^{-c_0}]\}/(1 - t)$; see [10].