On the Convergence of Simulated Annealing and Random-Start Local Search Algorithms for Combinatorial Optimization Problems*

Patrick Jaillet‡

December 1986

Abstract

In this technical note, we show that, for any given combinatorial optimization problem, and under very general conditions, a "repeated-random-start" local search algorithm is superior to a generalized version of the so-called "simulated annealing" algorithm, in the sense that, beyond some common finite running time threshold, the repeated-random-start local search algorithm's probability of finding an optimal solution is always larger than the corresponding simulated annealing algorithm's probability.

Keywords: Randomized Algorithms, Generalized Simulated Annealing, Random-Start Local Search, Combinatorial Optimization

*Working paper, CERMA-86-12-3

‡Mathematics Department, ENPC, Paris, France. Supported in part by NSF/CNRS

1 Introduction

Consider a generic combinatorial optimization problem $\mathcal{CO}$, i.e. a finite set Ω of feasible solutions and a function value $\phi : \Omega \mapsto R$, so that the problem is to find $\omega_{opt} \in \Omega$ such that $\phi(\omega_{opt}) \leq \phi(\omega)$ for all $\omega \in \Omega$.

We assume that Ω is equipped with a neighborhood structure, i.e. a mapping $\mathcal{N} : \Omega \mapsto 2^{\Omega}$, such that for all $\omega \in \Omega$, $\mathcal{N}(\omega)$ represents the set of feasible solutions that can be reached from ω in one "move". The action of going from one feasible solution to one of its neighbors will be called a neighbor move. We assume that the neighborhood structure is complete in the sense that one can go from any feasible solution to any other by following successive neighbor moves.

In graph theoretic terms, we can represent each feasible solution by a node and each neighbor move by an arc joining the two nodes. Successive neighbor moves will then correspond to paths in the graph.

An "only-down" path will refer to a path along which the function value is non-increasing. More precisely $\{i_1, i_2, \ldots, i_k\}$ is an "only-down" path if $\phi(i_1) \geq \phi(i_2) \geq \cdots \geq \phi(i_k)$. Define:

$\Omega^* = \{$feasible solutions from which all "only-down" paths end in a global minimal$\}$,

$\Omega^a = \{$feasible solutions from which there exists at least one "only-down" path ending in a global minimal$\}$.

We assume that $0 < |\Omega^*| = n^* \leq |\Omega^a| = n^a < |\Omega| = n$ in order to avoid trivial cases.

Finally let d be the maximum degree in the graph, and let h be the maximum span of the function ϕ on Ω, i.e. $h = \max_{\omega,\omega' \in \Omega}\{\phi(\omega) - \phi(\omega')\}$.

Let us now describe the two randomized algorithms for this combinatorial optimization problem.

1.1 Repeated-random-start local search

This algorithm is an adaptation of a classical local search algorithm which repeatedly starts from a random feasible solution and follows an "only-down" path toward a local minimal. The algorithm specifies the number of such repeated starts and chooses the best of the local minimal thus generated. More precisely the Repeated-Random-Start ($\mathcal{RRS}$) local search algorithm is defined by the following family of algorithms (parameterized by k, the a priori total number of random starts):

procedure $\mathcal{RRS}$ (k) ;
begin
$j := 1$;
while $j \leq k$ **do**
    pick a random solution $\omega(j) \in \Omega$;
    while there exists $\omega' \in \mathcal{N}(\omega(j))$ such that $\phi(\omega') < \phi(\omega(j))$ **do**
        $\omega(j) := \omega'$;
    $j := j + 1$;
return $\omega(i)$ such that $\phi(\omega(i)) = \min_{1 \leq j \leq k} \phi(\omega(j))$;
end.

Note that this algorithm is quite general and that a specific implementation would require a precise mechanism on how the step

"**while** there exists $\omega' \in \mathcal{N}(\omega(j))$ such that $\phi(\omega') < \phi(\omega(j))$ **do**"

is actually done. Two popular methods are either to (i) search the entire neighborhood and then move to the solution that decreases ϕ the most, or (ii) search the neighborhood and move to the first solution that decreases ϕ. For our analysis this specific choice is however irrelevant.

1.2 Generalized simulated annealing

For an introduction to the simulated annealing concepts in combinatorial optimization, we refer the reader to [1, 8]. Other fundamental developments on the use of simulated annealing in discrete and continuous optimization include [2, 3, 4, 5, 6, 7, 9].

In this paper we introduce a very general simulated annealing scheme. In order to describe it we need to specify a "cooling schedule":

$$(T_1, k_1), (T_2, k_2), \ldots, (T_i, k_i), \ldots,$$

where $T_1, T_2, \cdots, T_i, \cdots$, is a sequence of parameters (also called a "temperature" schedule), and where $k_1, k_2, \ldots, k_i, \ldots$ is a sequence of finite positive integers such that for each i, k_i represents the number of attempted moves done at temperature T_i. We also need to specify the probability $p(\omega, \omega', T_i)$ of moving from ω to $\omega' \in \mathcal{N}(\omega)$ at temperature T_i. The only condition we impose on these probabilities is as follows:

> Let $q_i = \inf_\omega\{1 - \sum_{\omega' \in \mathcal{N}'(\omega)} p(\omega, \omega', T_i)\}$, where $\mathcal{N}'(\omega) = \{\omega' \in \mathcal{N}(\omega) : \phi(\omega') > \phi(\omega)\}$. Then $\lim_{i\to\infty} q_i = 1$.

Note that q_i represents a lower bound on the probability of "not going up" from any solution at temperature T_i. This technical condition is essential for the simulated annealing concept, and is necessary (although not sufficient) for the convergence in probability of the algorithm to an optimal solution (in the sense that the random Markov chain produced by the algorithm will converge in distribution to a stationary probability measure π such that $\pi(\omega) > 0$ if and only if ω is a global minimal) . Note that the classical simulated annealing choice $p(\omega, \omega', T_i) = \frac{1}{|\mathcal{N}(\omega)|} e^{-\frac{[\phi(\omega') - \phi(\omega)]^+}{T_i}}$, with $T_i = c/\log(i+1)$ and c a sufficiently large constant (see e.g. [7]), verifies the condition stated above.

The Generalized Simulated Annealing ($\mathcal{GSA}$) algorithm can now be described by the following family of algorithms (parameterized by u and $(T_i, k_i)_{1\le i\le u}$):

procedure $\mathcal{GSA}$ $((T_i, k_i)_{1 \leq i \leq u})$;

begin

pick a random solution $\omega \in \Omega$;

$j := 1$;

while $j \leq u$ **do**

$i := 1$;

while $i \leq k_j$ **do**

$\omega := \omega' \in \mathcal{N}(\omega)$ with probability $p(\omega, \omega', T_j)$;

$i := i + 1$;

$j := j + 1$;

return ω ;

end.

1.3 Notions of elementary step and success

We will assume that for both algorithms it takes one elementary step to evaluate (and possibly move to) a neighbor solution from a given feasible solution, and that it takes r such elementary steps to get a random initial feasible solution.

The algorithms will said to succeed in less than N elementary steps if they go through a global minimal solution at least once in N elementary steps.

We will moreover consider a "super power" version of the Generalized Simulated Annealing which will be guaranteed to directly find an optimal solution as soon as it reaches a solution in Ω^a. In other words, this super algorithm, through "uncommon clairvoyance", is guaranteed to follow an only-down path toward global minimal as soon as it reaches a solution for which there is at least such a path.

2 Analysis of the algorithms

Let $k(N)$ and $u(N)$ be choices of the parameters in the previous family of algorithms so that both $\mathcal{RRS}(k(N))(\equiv \mathcal{A}_1(N))$ and $\mathcal{GSA}((T_i, k_i)_{1\leq i\leq u(N)}(\equiv \mathcal{A}_2(N))$ are guaranteed to run in at least N elementary steps.

It is not too difficult to see that $k(N) = \lceil N/r \rceil$ and that $u(N) = \min\{u : \sum_{i=1}^{u} k_i \geq N - r\}$ would be adequate choices.

We can now state our main results:

Lemma 1 *For any given N, we have:*

$$P(\mathcal{A}_1(N) \text{ succeeds in less than } N \text{ steps}) \geq 1 - (1 - \frac{n^*}{n})^{\lfloor \frac{N}{r+hd} \rfloor} \tag{1}$$

Proof: In N elementary steps, the algorithm will consider at least $\lfloor \frac{N}{r+hd} \rfloor$ different random starting solutions. Indeed each random starting solution and subsequent only-down path can consume up to $r + hd$ elementary steps. Now, if at least one of these random starting solutions is in Ω^*, then the algorithm would succeed. The probability of having no starting solution in Ω^* in $\lfloor \frac{N}{r+hd} \rfloor$ independent trials is $(1 - \frac{n^*}{n})^{\lfloor \frac{N}{r+hd} \rfloor}$. $\square$

Lemma 2 *For any given N, we have:*

$$P(\mathcal{A}_2(N) \text{ succeeds in less than } N \text{ steps}) \leq 1 - (1 - \frac{n^a}{n}) \prod_{j=1}^{u(N)} q_j^{k_j} \tag{2}$$

Proof: If the initial random solution is not in Ω^a, and if all subsequent moves are not going up then the super version of this algorithm would not succeed. The probability of the second condition is bounded from below by $\prod_{j=1}^{u(N)} q_j^{k_j}$. $\square$

Theorem 1 *There exists N_0 such that for all $N \geq N_0$:*

$$P(\mathcal{A}_1(N) \text{ succeeds in less than } N \text{ steps}) \geq P(\mathcal{A}_2(N) \text{ succeeds in less than } N \text{ steps}) \tag{3}$$

Proof: Obvious from the previous two lemmas and from the condition imposed on q_i. Indeed this condition implies that there exists j_0 large enough such that for all $j \geq j_0$:

$$q_j > (1 - \frac{n^*}{n})^{1/(r+hd)}. \tag{4}$$

Let then $q^* = \inf_{j \geq j_0} q_j$, $N_1 = r + \sum_{j=1}^{j_0-1} k_j$, and $\alpha = (1 - \frac{n^a}{n}) \prod_{j=1}^{j_0-1} q_j^{k_j}$. From (4), one can always find N_2 large enough such that for all $N \geq N_2$ we have:

$$\left(\frac{(1 - \frac{n^*}{n})^{1/(r+hd)}}{q^*} \right)^N \leq \alpha (1 - \frac{n^*}{n})^{1 - N_1/(r+hd)}, \tag{5}$$

which implies that

$$(1 - \frac{n^*}{n})^{\lfloor \frac{N+N_1}{r+hd} \rfloor} \leq (1 - \frac{n^*}{n})^{\frac{N+N_1}{r+hd} - 1} \leq \alpha (q^*)^N \leq \alpha q_m^k \prod_{j=j_0}^{m-1} q_j^{k_j}, \tag{6}$$

where m and k are defined by $\sum_{j=j_0}^{m-1} k_j + k = N$, and $0 \leq k < k_m$. By taking $N_0 = N_1 + N_2$, we finally get the desired result. $\square$

References

[1] V. Cerny. 1982. A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, preprint, Inst. of Phys. and Biophys., Comenius Univ., Bratislava.

[2] S. Geman et D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images, *IEEE Trans. PAMI*, 6, 721-741.

[3] S. Geman et C.R. Hwang. 1986. Diffusions for global optimization, *SIAM J. Cntrl. Opt.*, 24, 1031-1043.

[4] B. Gidas. 1985. Nonstationary markov chains and convergence of the annealing algorithm, *J. Stat. Phys.*, 39, 73-131.

[5] B. Gidas. 1985. Global optimization via the langevin equation, *Proc. IEEE Conf. Dec. and Cntrl.*

[6] B. Hajek. 1985. Cooling schedules for optimal annealing, preprint, Dep. Elec. Eng. , Univ. of Illinois, Champaign-Urbana.

[7] B. Hajek. 1985. A tutorial survey of theory and applications of simulated annealing, *Proc. IEEE Conf. Dec. and Cntrl.*

[8] S. Kirkpatrick, C.D. Gelatt et M. Vecchi. 1983. Optimization by simulated annealing, *Science*, 220, 621-680.

[9] J. Tsitsklis. 1985. Markov chains with rare transitions and simulated annealing, preprint, Dep. Elec. Eng., MIT, Cambridge.