Local Search for Statistical Counting

Olivier Bailleux

CRIL
Université d'Artois
rue de l'Université SP 16
F-62307 Lens Cedex, France
bailleux@cril.univ-artois.fr

Abstract

In this paper, statistical counting is introduced in the context of stochastic local search. From a sample of trajectories by independent local search computations, it is shown that interesting statistical information can be actually extracted about the search space, most notably an unbiased estimate of the number of solutions. Computational results for random #SAT instances are provided.

Introduction

The stochastic local search paradigm covers a large number of techniques that use a neighborhood relation and a so-called fitness function for exploring a search space. Typically, these techniques are used to address hard optimization and search problems (Selman & Kautz 1993) that cannot be solved by means of more standard methods. For example, some stochastic local search algorithms specialized for boolean satisfiability testing can solve positive instances of the boolean satisfiability problem (SAT) that are intractable for all known systematic approaches.

Generally speaking, a local search process tries to find out a solution by exploring a search landscape, i.e. a neighborhood graph labeled with a fitness (or cost) function.

The performance of a stochastic local search system depends on both the landscape and the search strategy. One of the simplest strategies for maximization problems is known as hill climbing. It consists in drawing each new configuration uniformly from the set of the current configuration neighbors that increase the fitness function. The drawback of this strategy is that the search stops at local extrema. The greedy search is a variant that choose each new configuration from those that maximize the fitness function (even if the fitness decreases). More sophisticated techniques, like simulated annealing (van Laarhoven & Aarts 1987), introduce a noise in order to escape from local extrema. Another technique, known as tabu search(Mazure, Saïs, & Grégoire 1997), deals with local extrema by storing the most recently visited configurations in a tabu list in order to avoid them during the next steps. Because they can be used on all discrete optimization and search problems, the strategies like tabu, simulated annealing or greedy search are called *meta-heuristics*, in contrast with specialized local search algorithms that use specific problem features, like WSAT algorithm (Selman, Kautz, & Cohen 1994) for satisfiability testing.

Beyond assessing the efficiency of local search algorithms, some experimental studies have been conducted so far, relating the behavior of local search processes to topological properties of search landscapes. For example, see (Frank, Cheeseman, & Stutz 1997) for a study of landscapes related to random 3SAT instances.

Let us now recall the class #P and the concept of counting Turing machine (CTM) which was introduced by Valiant in (Valiant 1979). A CTM is a non deterministic Turing machine with an auxiliary device that "magically" outputs the number of accepting computations induced by the input. The time complexity of a CTM is defined as the time complexity of the related non deterministic Turing machine (with no auxiliary device). So, a CTM exhibits a polynomial time complexity iff there is a polynomial p such that the longest accepting computation induced by the set of all the inputs of size n takes at most $p(n)$ steps. The class $\#P$ is the set of functions that can be computed in polynomial time by a CTM. #SAT, the problem of counting the number of satisfying assignments of a boolean formula that is expressed in conjunctive normal form, is $\#P$-Complete. The restriction of #SAT to the boolean formulae with only positive literals (#MONOTONE-SAT), and even to the monotone formulae with only binary clauses (#2-MONOTONE-SAT), remains $\#P$-Complete.

The relation $PH \subseteq P^{\#P}$, where PH denotes the polynomial hierarchy (Garey & Johnson 1979) and $P^{\#P}$ denotes the set of problems that can be solved in deterministic polynomial time with a $\#P$ oracle, follows directly from Toda's theorem (Toda 1989). This relation shows the hardness of #P-Complete problems like #SAT, #MONOTONE-SAT, #2SAT. In (Roth 1996), D. Roth shows that for any constant $\varepsilon > 0$, approximating the number of satisfying assignments of a 2-MONOTONE boolean formula of n variables within $2^{n^{1-\varepsilon}}$ is NP-Hard. So, even approximate resolution of #2-MONOTONE-SAT is intractable under the assumption $P \neq NP$.

In (Bailleux & Chabrier 1996) and (Bailleux & Chabrier 1997), it is proposed to use a statistical exploration of a search tree for approximate resolution of numbering problems.

In this paper, a concept of statistical counting is introduced in the context of stochastic local search. From a sample of trajectories by independent local search computations, it is shown that interesting statistical information can be actually extracted about the search space, most notably an unbiased estimate of the number of solutions.

This approach can be used for the approximate resolution of hard numbering problems (typically #P-Hard problems). Despite a huge variance, a statistical lower bound of the solution can be obtain with a given probability of error.

The proposed method does not advance the state of the art in terms of efficiency or quality of estimates, at least for random #SAT instances. Our motivation is rather to show that, from a sample of solutions obtained through stochastic local search, global information about the search landscape can be derived. As a result, the local search paradigm is not necessarily restricted to satisfiability testing or solution searching, but could be *potentially* extended to statistical counting and search landscapes analysis.

The paper is organized as follows. In a first section, a lemma is presented, which makes the connection between the size of two partitions of a finite set and the mathematical expectation of a random variable defined on this set. Then the application of this lemma for statistical counting by making two partitions of the set of possible trajectories of a stochastic local search process is described. In a second section, computational results from random #SAT instances are provided.

Formal framework

Principle

In this section, a lemma is presented, which makes the connection between the sizes of two partitions E and F of a finite set Ω and the mathematical expectation of a random variable X defined on Ω.

Let W be a function that maps Ω into $[0, 1]$ s.t. for each $f \in F$, $\sum_{t \in f} W(t) = 1$.

Let V be a function that maps Ω into $]0, 1]$ s.t. for each $e \in E$, $\sum_{t \in e} V(t) = 1$.

Let X be a random variable on Ω s.t. for all $t \in \Omega$, $X(t) = \frac{W(t)}{V(t)}$ and the probability of t is $P(t) = \frac{1}{|E|} V(t)$.

Let $E[X]$ denote the mathematical expectation of X.

Lemma 1 $E[X] = \frac{|F|}{|E|}$.

Proof :

$$\begin{aligned} E[X] &= \sum_{t \in \Omega} P(t) X(t) \\ &= \sum_{t \in \Omega} \frac{1}{|E|} W(t) \\ &= \frac{1}{|E|} \sum_{t \in \Omega} W(t) \\ &= \frac{1}{|E|} \sum_{f \in F} \sum_{t \in f} W(t) = \frac{|F|}{|E|} \end{aligned}$$

□

Suppose that we know $|E|$ and are able to compute $V(t)$ and $W(t)$ for any $t \in \Omega$. Thanks to Lemma 1, we can estimate $|F|$ in the following manner: Let t be drawn from Ω according to the probability $P = \frac{1}{|E|} V$. Then $|E| \frac{W(t)}{V(t)}$ is an unbiased estimate of $|F|$.

Search landscape

A *search landscape* is the information a stochastic local search algorithm needs to address a problem instance. In the simplest case, a search landscape is a labeled graph (U, R, g) where U is the search space (a finite set of configurations), $R \subseteq U \times U$ is a neighborhood relation and g is a fitness (or cost) function which maps U into an ordered set (typically the set of integer numbers). When dealing with optimization problems, the fitness function is part of the problem itself. For search problems, a fitness (resp. cost) function that is maximal (resp. minimal) for solutions, only, must be pointed out.

Let us define a *trajectory* as a sequence of configurations $\alpha_1, ..., \alpha_n$, s.t. for each $i \in 2..n$, $(\alpha_{i-1}, \alpha_i) \in R$ (i.e. α_i is in the neighborhood of α_{i-1}). Then a local search process is simply a process that follows a trajectory in a search landscape.

Application to hill climbing

The main idea is to make two partitions E and F of the set Ω of all possible trajectories of a stochastic Hill Climbing process (HC).

For each $x \in U$, let $h(x)$ (resp. $l(x)$) denote the set of neighbors of x that increase (resp. decrease) the fitness function.

The process HC starts from a configuration uniformly drawn in the search space U. Each new configuration is drawn uniformly from the set $h(x)$, where x is the current configuration, until $h(x) = \emptyset$. Each run stops either at a local or a global extremum.

The first partition E consists in bringing together all the trajectories starting from the same configuration. Then $|E| = |U|$.

The second partition F consists in bringing together all the trajectories ending at the same global extremum. The trajectories ending at a *local* extremum are arbitrary dispatched in the sets assigned to the global extrema, in such a way that $|F|$ equals the number of global extrema.

Let $first(t)$ (resp. $last(t)$) denote the first configuration (resp. the last configuration) of a trajectory t. For each $t \in \Omega$, let $V(t)$ denote the probability that HC returns the trajectory t, given that the search starts from $first(t)$. Notice that $\frac{V}{|E|}$ is a probability on Ω and for each $e \in E$, $\sum_{t \in e} V(t) = 1$.

For a trajectory $t = (\alpha_1, \ldots, \alpha_n)$, $V(t)$ can be computed as follows:

$$\begin{cases} \text{if } n = 1 \text{ then } V(t) = 1 \\ \text{else } V(t) = \prod_{i=1}^{n-1} \frac{1}{|h(\alpha_i)|} \end{cases}$$

In order to take advantage from lemma 1, we need a function W s.t. for each $f \in F$, $\sum_{t \in f} W(t) = 1$. It must be possible to compute $W(t)$ for any $t \in \Omega$, whenever we want W to be actually useful.

Let PV be a "virtual" stochastic process (i.e. a process which will never run), starting from a given global extremum. At each step, PV considers the set $l(x)$, where x is the current configuration. When $l(x) = \emptyset$, the search stops. When $l(x) \neq \emptyset$, the search stops with probability Q, or continues from a new configuration, uniformly drawn in $l(x)$, with probability $1 - Q$.

We chose the following function W: For each $t = (\alpha_1, \ldots, \alpha_n) \in \Omega$, if t ends at a local extrema then $W(t) = 0$, else $W(t)$ is the probability that PV returns $\tilde{t} = (\alpha_n, \ldots, \alpha_1)$, given that the search starts from α_n.

Let S be the set of global extrema. For a trajectory $t = (\alpha_1, \ldots, \alpha_n)$, $W(t)$ can be computed as follows:

$$\begin{cases} \text{if } last(t) \notin S \text{ then } W(t) = 0 \\ \text{else } W(t) = \lambda(Q, t) \end{cases}$$

Where $Q \in [0, 1]$ is a constant and $\lambda(Q, t)$, is defined as:

$$\begin{cases} \text{if } n = 1 \text{ and } |l(\alpha_1)| \neq 0 \text{ then } \lambda(Q, t) = Q \\ \text{if } n = 1 \text{ and } |l(\alpha_1)| = 0 \text{ then } \lambda(Q, t) = 1 \\ \text{if } n > 1 \text{ then } \lambda(Q, t) = \frac{1-Q}{|l(\alpha_n)|} \lambda(Q, (\alpha_1, \ldots, \alpha_{n-1})) \end{cases}$$

Yet, assuming we know $|U|$, all the conditions are right for estimating $|S|$ from a sample $(t_1, \ldots, t_m)$ of trajectories achieved by m independent computations of HC. The value μ defined as follows is an unbiased estimate of $|S|$:

$$\mu = |U| \frac{1}{m} \sum_{i=1}^{m} \frac{W(t_i)}{V(t_i)}$$

Using Markov's inequality, we can compute a lower bound of $|S|$ with a given probability of error. Because the random variable X cannot take a negative value, we have: $P(X > \gamma E[X]) \leq \frac{1}{\gamma}$ for any constant $\gamma > 0$.

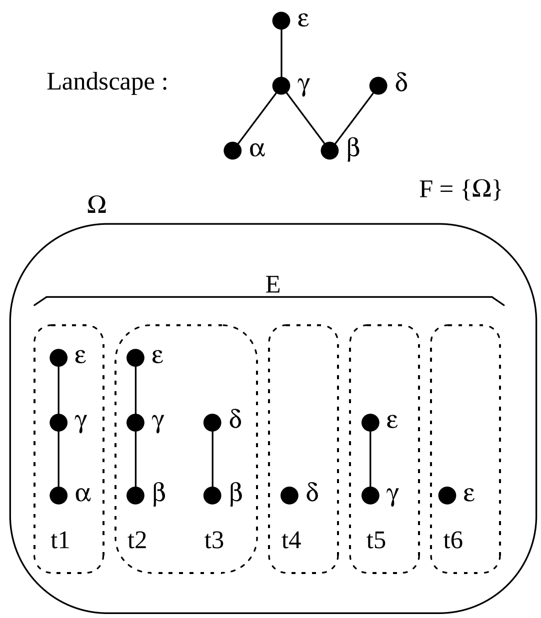

Figure 1: The set Ω and the partitions E and F for a simple landscape.

So, if $\mu_1, \ldots, \mu_n$ is a sequence of unbiased estimates of $|S|$, then we can claim that $\frac{1}{\gamma} \min\{\mu_1, \ldots, \mu_n\}$ is lower than $|S|$ with probability of error at most equals to $1/\gamma^n$.

Notice that if $Q = 1$ the only trajectories for which $X > 0$ are the ones that start directly from a global extremum ($first(t) = last(t) \in S$). So, the method collapses to the estimate of $|S|$ by uniform sampling from the search space. If $0 < Q < 1$, all the trajectories ending with a global extremum are taken into account. At last, if $Q = 0$, only the trajectories t for which $|l(first(t))| = 0$ are taken into account.

We suppose (and we show experimentally in the next section) that for a given landscape, quality of estimates depends on the value of Q. Actually, we have no way to determine the optimal value of Q for landscapes related to #P-complete (or even simpler) problems.

As an example, Figure 1 presents, for a simple landscape, the set Ω and the partitions E and F. For $Q = 0.5$, the following values are assigned by V (resp. W) to the trajectories $t_1, ..., t_6$:

	t_1	t_2	t_3	t_4	t_5	t_6
V	1	1/2	1/2	1	1	1
W	1/8	1/8	0	0	1/4	1/2

It follows that $E[X] = 1/5 = \frac{|F|}{|E|}$.

Computational results

Implementation for SAT

The Boolean Satisfiability Problem (SAT) is the problem to decide whether an equation $\phi = 1$ – where ϕ is a boolean formula in conjunctive normal form – has at least one solution (Cook 1971). So, a SAT instance is a formula $c_1 \wedge \ldots \wedge c_k$ where each c_i is a clause, i.e. a disjunction $v_1^{\varepsilon_1} \vee \ldots \vee v_r^{\varepsilon_r}$, where each literal v_i^0 (resp. v_i^1) denotes the complemented form (resp. the direct form) of the variable v_i.

The rSAT problem (i.e. the restriction of SAT to instances with r literals per clauses) is NP-Complete for $r > 2$. Thereby, 3SAT, one of the simplest NP-Complete problem, is often used to evaluate the solving algorithms efficiency.

The #rSAT problem (the numbering problem related to rSAT) is the problem to determine the number of solutions of an equation $\phi = 1$, where ϕ is a CNF boolean formula with at most r literals per clause. It is #P-complete for $r > 1$, even if ϕ is monotone.

In our experimentations, the landscape related to a formula ϕ on n variables is $L_\phi = (U, R, g)$, where $U = \{0,1\}^n$ and for all $x, y \in U$,

- $(x, y) \in R$ if and only if the Hamming distance $H(x, y) = 1$,
- $g(x)$ is the number of clauses satisfied by x.

Experimentations on tractable random #SAT instances

In this section, we give results obtained from instances for which systematic counting methods are usable. This allows us to compare each estimate with the related exact number of solutions.

Let us define a *reduced* clause as a clause each variable of which occurs at most one time.

Our benchmarks include 3SAT instances each clause of which is drawn uniformly from the set of all possible reduced clauses, for a given number of variables.

For tractability reasons, random 3SAT instances of at most 50 variables were addressed.

In the sequel, k/n will denote the ratio of the number of clauses by the number of variables of a 3SAT instance.

Let us define the *relative error* of an estimate μ, achieved from a sample of trajectories related to an instance of s solutions, as $|\mu - s| / s$.

Because a few estimates could exhibit a relative error much higher than others, we chose to use the average on the 15-85 percentile range (i.e. trimmed mean) of relative errors for evaluating the quality of a sample of estimates.

According to the parameter Q, Figure 2 (resp. 3) shows the relative error for random 3SAT instances of 30 (resp. 50) variables with $k/n = 2$ and $k/n = 3$ (500 measures per point, 10000 trajectories per estimate).

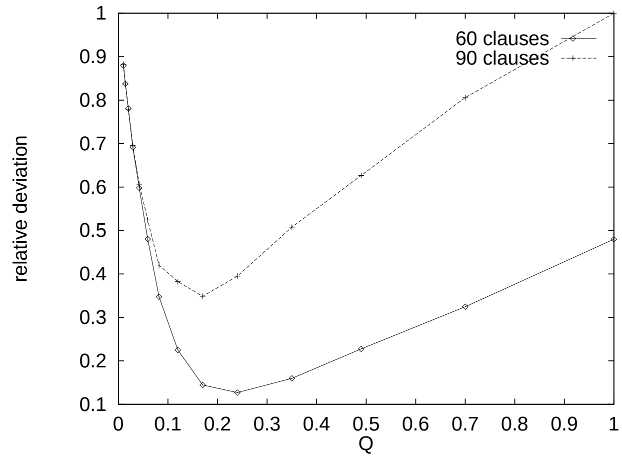

Figure 2: Random 3SAT instances, 30 variables. Relative error of estimates (Average on 15-85 percentile range, 500 mesures per point, 10000 trajectories per estimate).

Notice that in terms of exact counting, instances with $k/n = 2$ or 3 are much more difficult than instances with $k/n = 4.25$, usually used for evaluating search algorithms. For example, counting the solutions of random 50 variables 3SAT instances with a Davis and Putman procedure and Jeroslow's branching rule (Jeroslow & Wang 1990) requires on average 10^6 branches to be developed for $k/n = 2$, 10^4 for $k/n = 3$ and 200 for $k/n = 4.25$.

Empirically, the quality of estimates depends on the Q parameter. Local search based estimates are better than estimates based on uniform sampling, given that the best estimates are observed for $Q < 1$.

Experimentations on intractable random #SAT instances

In this section, we are interested in randomly generated instances of 200 variables for which we do not know any reference of usable exact counting method.

For several values of the ratio k/n, Figure 4 gives the average of estimated number of solutions of random 3SAT instances obtained on the one hand using our local search based method, on the other hand using uniform sampling from the search space. Theses results are compared with the theoretical expectation of the number of solutions (Simon & Dubois 1989), that is :

$$2^n \left(1 - \frac{1}{2^r}\right)^k$$

where $r = 3$ is the number of literals per clause, n is the number of variables and k is the number of clauses.

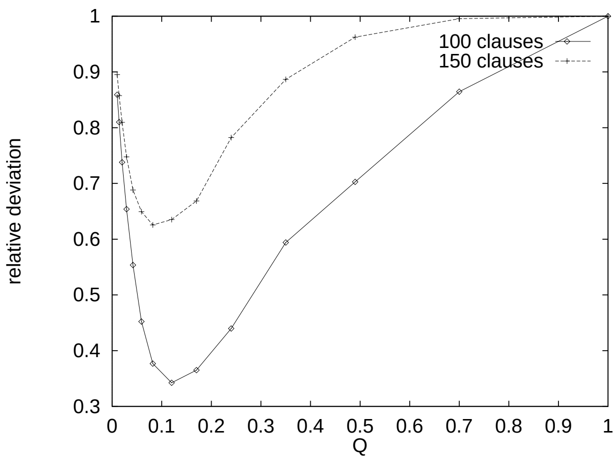

Figure 3: Random 3SAT instances, 50 variables. Relative error of estimates (Average on 15-85 percentile range, 500 mesures per point, 10000 trajectories per estimate).

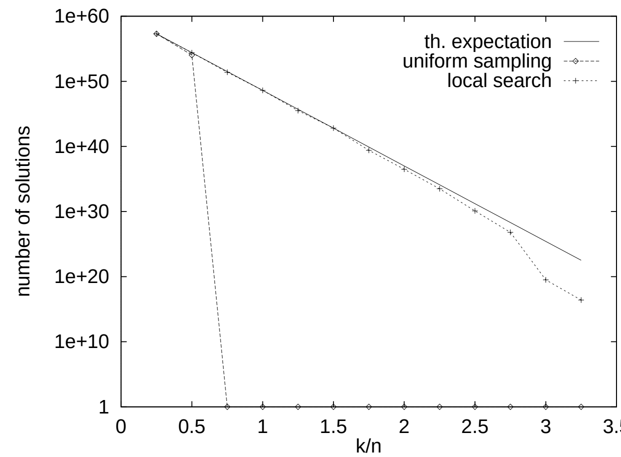

Figure 4: Random 3SAT instances of 200 variables. Mathematical expectation of the number of solutions, local search based estimates and estimates by uniform sampling. 300 mesures per point, 10000 trajectories (resp. configurations) per mesure.

For each value of k/n, 300 estimates were computed. Each estimate was obtained by sampling 10^4 trajectories or 10^4 configurations, respectively.

Either way, the quality of estimates is getting worse when the number of solutions decreases, but this phenomenon occurs much more quickly with uniform sampling. Although these results are slightly worse than those obtained with a randomized David and Putman procedure, especially for $k/n > 3$ (Bailleux & Chabrier 1997), local search based sampling appears to be a better candidate than uniform sampling.

The fact that bad estimates are often *below* the theoretical expectation is typical of positive random variables with high dispersion.

Conclusion

In this paper, a stochastic local search based technique for estimating the number of global extrema in a search landscape has been proposed. It is based on a statistical analysis of a sample of trajectories achieved by a stochastic hill climbing process. It allows us to derive information on landscapes the density of global extrema of which is too low for an estimation by uniform sampling from the search space.

The estimates are unbiased but the variance, which can be very high, is unknown in the general case. So, we cannot obtain a confidence interval but a lower bound with a given (arbitrary low) probability of error.

Our experimentations on random #SAT instances show that there are landscapes for which stochastic local search sampling gives much better estimates than uniform sampling from the search space.

Although the proposed approach does not outperform existing statistical methods, at least for random 3SAT instances, it presents a theoretical and prospective interest.

As a research perspective, we plan to extend this approach to the estimation of other parameters of search landscapes, like the density of local extrema and the number of points with a given fitness. Another perspective is to extend the method in order to cover more efficient local search algorithms like GSAT or WSAT.

Acknowledgments

This work has been supported in part by the Ganymede II project of the ”Contrat de plan Etat/Nord-Pas-de-Calais”. Many thanks to Jean-Jacques Chabrier (Lirsia, Dijon, France), Yacine Boufkhad and Eric Grégoire (CRIL, Lens, France) for useful discussions and comments on this paper.

References

Bailleux, O., and Chabrier, J.-J. 1996. Approximate resolution of hard numbering problems. In *Proceedings of AAAI96*, 169–174.

Bailleux, O., and Chabrier, J.-J. 1997. Counting by statistics on search trees: Application to constraint satisfaction problems. *Intelligent Data Analysis* 1(4). http://www.elsevier.com/locate/ida.

Cook, S. 1971. The complexity of theorem-proving procedures. In *Proc. 3rd Ann. ACM Symp. on Theory of Computing*, 151–158.

Frank, J.; Cheeseman, P.; and Stutz, J. 1997. When gravity fails : Local search topology. *Journal of Artificial Intelligence Research* 7:249–281.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability*. W. H. Freeman and Compagny.

Jeroslow, R., and Wang, J. 1990. Solving propositional satisfiability problems. *Ann. Math. AI* 1:167–187.

Mazure, B.; Saïs, L.; and Grégoire, E. 1997. Tabu search for sat. In *Proceedings of AAAI97*, 281–285.

Roth, D. 1996. On the hardness of approximate reasoning. *Artificial Intelligence* 82:273–302.

Selman, B., and Kautz, H. 1993. An empirical study of greedy local search algorithms for satisfiability testing. In *Proceedings of the 11th National Conference on Artificial Intelligence*.

Selman, B.; Kautz, H.; and Cohen, B. 1994. Noises strategies for improving local search. In *Proceedings of AAAI94*, 337–343.

Simon, J. C., and Dubois, O. 1989. Number of solutions of satisfiability instances : application to knowledge bases. *International Journal of Pattern Recognition and Artificial intelligence* 3(1):53–65.

Toda, S. 1989. On the computational power of pp and +p. In *Proc. 30th IEEE Symp. on the Foundations of Computer Science*, 514–519.

Valiant, L. 1979. The complexity of enumeration and reliability problems. *SIAM J. Comput.* 8(3):410–421.

van Laarhoven, P. J. M., and Aarts, E. H. L. 1987. *Simulated annealing : Theory and Applications*. Kluwer Academic Publisher.