Random walk and parallelism in local search

Andrea Roli* Maria J. Blesa† Christian Blum†

*Dipartimento di Scienze, Università "G.D'Annunzio"
Viale Pindaro 42 – I-65127 Pescara, Italia
a.roli@unich.it

†ALBCOM, LSI, Universitat Politècnica de Catalunya
Jordi Girona 1-3, Campus Nord – E08034 Barcelona, Spain
{cblum,mjblesa}@lsi.upc.edu

1 Introduction

The introduction of randomness in local search is a common way of improving the average performance, in terms of solution quality.[1] A typical example is the use of randomness to choose for a move, instead of using a heuristic criterion. In that context, in [8, 10, 5] it has been shown that the introduction of *noise* (i.e., randomness) into local search for tackling the Satisfiability Problem is particularly effective. Another way to boosting local search is achieved by applying more than one move at a time, i.e., by performing *parallel moves*[2]. This technique has its roots in the so called *criticality and parallelism* phenomenon [7]. Noise and parallelism undergo an analogous behavior when applied to local search. They both involve a parameter depending on which local search performance exhibits a *bad–good–bad* behavior, as graphically shown in Fig.1. Local search performance increases as the amount of noise (resp. parallelism) increases, up to a point at which the performance starts to decrease.

In this work, we report and discuss preliminary experiments aimed at checking if the two techniques interfere when applied at the same time. We performed experiments on random instances of the Satisfiability Problem and we observed that the algorithm resulting from the combination of both multi-move and noise local search outperforms local search equipped only with either multi-move or noise.

This paper has the following organization. In Secs.2 and 3, respectively, we summarize the results achieved in the context of noise strategies and parallelism in local search. In Sec.4, we present the algorithm resulting from the combination of multi-move and noise and we show experimental results on random 3-SAT instances. Finally, we conclude with a discussion and an outline of future work.

[1]In the following we will use the expression *solution quality* referring to the objective function value. In case of a minimization problem, the lower the objective value, the higher the solution quality.

[2]Throughout the paper, we will use the expressions 'parallel moves' and 'multi-move' interchangeably.

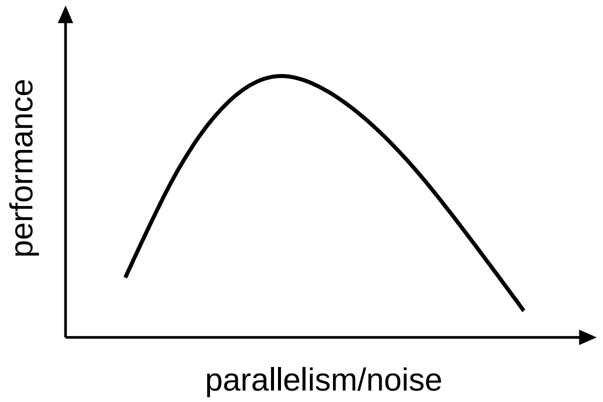

Figure 1: Typical performance behavior of local search as a function of a parameter that, in turns, represents the amount of parallelization (i.e., number of simultaneous moves) and noise.

2 Noise and local search

The introduction of a moderate amount of *noise* into local search is a common practice to achieve diversification [2]. The most adopted noise techniques in local search are random moves, wherein moves are chosen according to a uniform distribution[3]. A generalized class of noise algorithms is described in [3].

A very interesting study on the effect of noise on local search has been discussed in [8, 10]. In that work, it has been shown that the introduction of noise increases the effectiveness of local search, since it helps to escape from local optima. Furthermore, the quality of solutions found increases as noise increases, up to a critical value above which the performance decreases again. The algorithm considered in [8] is WalkSAT [14] combined with random walk, applied to the Satisfiability Problem (SAT). SAT belongs to the class of NP-complete problems [4] and can be stated as follows: given a set of clauses, each of which is the logical disjunction of $k > 2$ *literals* (a literal is a variable or its negation), we ask whether an assignment to the variables exists that satisfies all the clauses. WalkSAT is based on the principle of repair: It randomly chooses one unsatisfied clause and flips[4] one variable within it. There are some different heuristics for the choice of the variable to flip, that are discussed and studied also in [5]. When the variable is randomly chosen, we say that a random walk move is performed. It has been experimentally shown that low values of randomness do not significantly boost local search and high values of randomness can even deteriorating the algorithm performance. In between, there is an optimal value that enables the algorithm to reach an optimal performance. The actual optimal balance between heuristic and random choice of the variable to flip is instance dependent and it can be only experimentally estimated. The intuition behind this phenomenon is that low level of noise does not help local search to escape from local minima and too strong noise inhibits the search strategy, making the search degenerating to random search. Therefore, an intermediate level of randomness is the optimal trade-off between exploration and exploitation.

[3] In this paper we always refer to uniform distribution, excluding the use of other probabilistic distributions.
[4] A *flip* is a move performed on a binary variable, implemented by changing its value.

3 Multi-move local search

The phenomenon called *criticality and parallelism* has been observed in the context of local search algorithms applied to combinatorial optimization problems [7], where local search is modified by applying more than one local move in parallel. It has been shown that the effectiveness of these algorithms depends on the parallelism degree τ (number of simultaneous moves): If τ increases, the solution quality also increases up to a point (corresponding to τ_{opt}) at which it starts to decrease. It has also been shown that τ_{opt} is negatively correlated with the *connectivity* among variables of the problem: The higher the connectivity, the lower τ_{opt}.

A similar phenomenon has been discussed in [12, 13], where a multi-flip version of GSAT [15] has been applied to SAT and its maximization version MAXSAT instances. GSAT is a local search algorithm based on a greedy search strategy. In its basic version, it starts from a random assignment and looks for a satisfying assignment by moving from one state to another one in its neighborhood (defined as the set of states at Hamming distance equal to 1). Given a current state, the next state is chosen by flipping the variable that, if flipped, leads to the greatest increment of satisfied clauses. There is a very easy way to parallelize GSAT: the set of variables is divided in equal subsets. If n is the number of variables, the number of subsets corresponds to the parallelism degree τ and the cardinality of each subset is n/τ (except for one subset that contains $\tau + n \ mod \ \tau$ variables). The procedure obtained (thereinafter referred to as PGSAT), behaves as follows: at each iteration, the subsets are considered in parallel and the "best" variable for each of them is flipped. Therefore, after an iteration, τ variables will be flipped.

Previous results on this subject can be summarized as follows:

(i) there is a value of $\tau_{opt} \in \{1, \dots, n\}$ that leads PGSAT to achieve the best performance. The average error returned by PGSAT(τ_{opt}) is not higher than that of PGSAT(τ), for $\tau \in \{1, \dots, n\}$;

(ii) τ_{opt} is negatively correlated with the average node degree $\overline{q}$ of the *interaction* graph[5] associated to the SAT instance: the higher $\overline{q}$, the smaller τ_{opt}.

4 Combining parallel flips and random walk

In the previous sections, we have seen that multi-flip local search and noise strategy local search have a common behavior. In this work, we aim at checking what is the outcome of the combination of the two techniques. They may have either a negative or positive interference. In the first case, the combined technique is worse, or at most equal, than any single one. In the second case, the advantages of the two single techniques are effectively combined. Multi-flip and noise induce a complex dynamics in local search and researching on this issue has relevance for the understanding of the local search process. Therefore, the tests we perform in this work can be considered as an initial step in this direction.

[5]In the interaction graph [11], nodes correspond to variables and edges connect two variables if there exists a clause involving them.

Table 1: Comparison of the average error (along with std. dev.) and max error values returned by the algorithms (min error is 0 for all the algorithms). Averages are computed over 1000 runs. Parameter settings: $n = 20$: wp=0.6, τ=3; $n = 50$: wp=0.5, τ=5; $n = 100$: wp=0.5, τ=7; $n = 150$: wp=0.3, τ=12; $n = 200$: wp=0.6, τ=11; $n = 250$: wp=0.7, τ=12.

# of	GSAT		PGSAT		GWSAT		PGWSAT	
vars	ave. (std.dev)	max	ave. (std.dev)	max	ave. (std.dev)	max	ave. (std.dev)	max
20	0.751 (0.909)	3	0.751 (0.862)	3	0.090 (0.415)	2	0.006 (0.077)	1
50	2.146 (1.199)	7	1.364 (0.909)	4	1.217 (0.986)	5	0.388 (0.578)	2
100	3.688 (1.665)	10	2.292 (1.300)	7	2.205 (1.469)	7	1.007 (0.827)	4
150	3.600 (1.911)	11	1.620 (1.203)	6	1.800 (1.355)	8	0.278 (0.574)	4
200	5.736 (2.102)	14	3.231 (1.297)	8	3.378 (1.559)	10	1.712 (0.902)	7
250	4.506 (2.159)	12	1.828 (1.164)	7	2.005 (1.410)	8	0.677 (0.657)	5

Since in our previous experiments on multi-flip local search we used GSAT, we can easily add noise strategies by moving from GSAT to GWSAT, that is the combination of GSAT and random walk. With probability wp, instead of performing a GSAT move, an unsatisfied clause is randomly chosen and a variable within it is selected at random and flipped. The algorithm we implemented, PGWSAT, is the combination of PGSAT and random walk: with probability wp, a repairing move is performed and with probability $1 - wp$, a usual PGSAT move is taken. The optimal value for both wp (noise) and τ (parallelism) have been tuned as if the strategies were run independently. Then, we tested PGWSAT with that parameter setting on the benchmark instances. The algorithms GSAT, PGSAT, GWSAT and PGWSAT have been run on random 3-SAT instances retrieved from SATLIB [6]. The instances taken from SATLIB are located in the threshold region [1, 9] (i.e., $m/n \approx 4.3$, where n and m are respectively the number of variables and clauses) and are satisfiable. The algorithms has been stopped when a maximum number of non-improving moves was reached[6], i.e., until stagnation.

In Table 1, we reported the average error —evaluated as the number of unsatisfied clauses returned at the end of the run—, its standard deviation and maximal error values (all algorithms solved the instances at least once, so the minimal error is always 0). Averages are computed over 1000 runs on one instance from the benchmark. The results shown represent the typical behavior observed across the instances of the benchmark. We note that PGWSAT achieves an average solution quality notably higher than the solution quality of the other algorithms and it is also far more robust. Moreover, PGWSAT also outperforms GSAT, PGSAT and GWSAT in terms of success ratio (percentage of successful runs out of 1000) as shown in Table 2. Therefore, PGWSAT is also more effective than the other algorithms.

We can conclude that, at least for the benchmark we tackled, the combination of multi-flip and random walk is extremely effective and the two techniques show a synergistic positive effect. This can be explained by considering that multi-flip local search is usually able to find quickly areas of the search space where good solutions are located. These areas are effectively explored by the single-flip search performed by random walk.

[6]this value has been set to n

Table 2: Comparison of the success ratio of the algorithms. The percentage of success is evaluated over 1000 runs. Parameter settings are the same as for Table 1.

instance	Success ratio			
size	GSAT	PGSAT	GWSAT	PGWSAT
20	54.10%	54.10%	95.50%	95.50%
50	7.00%	16.40%	26.90%	66.00%
100	2.10%	2.70%	14.80%	24.90%
150	3.00%	13.30%	15.40%	73.10%
200	0.10%	2.00%	2.00%	8.60%
250	0.50%	8.40%	8.40%	37.80%

5 Conclusion and future work

In this work, we have investigated the effect of the combination of noise and parallelism (i.e., multi-flip moves) in local search. Noise and parallelism have a common behavior: Increasing noise (resp. parallelism) makes local search performance increasing up to a point at which it starts to decrease. The aim of the investigation was to check if the two techniques interfere. We implemented a variant of GSAT that incorporates both random walks and parallel flips (PGWSAT). We tested PGWSAT on hard random 3-SAT instances and we observed that, when the parameters are optimally tuned, PGWSAT strongly outperforms both PGSAT (the version only with parallel flips) and GWSAT (GSAT plus random walk).

Future work will be first focused on evaluating the algorithms on large structured SAT instances. Since results are extremely promising, we can also envision the use of the combination of parallel moves and noise to boost local search in different context than SAT. It is also very interesting to note that the parameters can be tuned independently, thus reducing the parameter search space.

Finally, it is important to notice that the algorithm we implemented can be interpreted as a multi-neighborhood local search. In fact, PGWSAT applies two kinds of moves: One with τ simultaneous flips (decided on the basis of a greedy criterion) and the other with a single flip (decided on the basis of a repair method). Therefore, in this perspective, it is not surprising that the two techniques have a combined positive effect.

Besides the possible applications of this result, this work also aims at investigating phenomena arising in the context of local search dynamics, still not completely understood. We believe that this research line is relevant in order to achieve a fruitful knowledge of local search process dynamics.

Acknowledgments

The authors thank Holger Hoos for suggesting to investigate the combination of Parallel GSAT and GWSAT.

References

[1] D. Achlioptas and C. Moore. The asymptotic order of the random k -SAT threshold. In *Proceedings of FOCS 2002*, pages 779–788, 2002.

[2] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003.

[3] I. Charon and O. Hudry. The noising methods: A generalization of some metaheuristics. *European Journal of Operational Research*, 135:86–101, 2001.

[4] M. R. Garey and D. S. Johnson. *Computers and intractability; a guide to the theory of NP-completeness.* W.H. Freeman, 1979.

[5] H.H. Hoos and T. Stützle. Towards a characterisation of the behaviour of stochastic local search algorithms for SAT. *Artificial Intelligence*, 112:213–232, 1999.

[6] H.H. Hoos and T. Stützle. SATLIB: An online resource for research on SAT. In I. Gent, H. van Maaren, and T. Walsh, editors, *SAT2000: Highlights of Satisfiability Research in the Year 2000*, pages 283–292. IOS Press, 2000.

[7] W. G. Macready and S. A. Kauffman A. G. Siapas. Criticality and parallelism in combinatorial optimization. *Science*, 271:56–59, January 1996.

[8] D. McAllester, B. Selman, and H. Kautz. Evidence for invariants in local search. In *Proc. of AAAI97*, pages 321–326, 1997. AAAI Press.

[9] D. G. Mitchell, B. Selman, and H. J. Levesque. Hard and easy distributions of SAT problems. In *Proc. of AAAI92*, pages 459–465. AAAI Press/MIT Press, July 1992.

[10] D. J. Patterson and H. Kautz. Auto-Walksat: A self-tuning implementation of WalkSAT. *Electronic Notes in Discrete Mathematics*, volume 9. Elsevier, 2001.

[11] I. Rish and R. Dechter. Resolution versus search: Two strategies for SAT. *J. Automated Reasoning*, 24:225–275, 2000.

[12] A. Roli. Criticality and parallelism in GSAT. *Electronic Notes in Discrete Mathematics*, volume 9. Elsevier, 2001.

[13] A. Roli and C. Blum. Critical Parallelization of Local Search for MAX–SAT. *Lecture Notes in Artificial Intelligence*, 2175:147–158. Springer-Verlag, 2001.

[14] B. Selman, H. A. Kautz, and B. Cohen. Noise strategies for local search. In *Proc.of AAAI94*, Seattle/WA, USA, pages 337–343, 1994.

[15] B. Selman, H. J. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *Proc. of AAAI92*, pages 440–446, 1992. AAAI Press.