A Scatter Search Variant to Solve max-SAT Problems

Dalila BOUGHACI[1], Habiba DRIAS[2], and Belaid BENHAMOU[3]

[1,2] Department of Computer Science, University of sciences and technology
BP.32- EL-ALIA Bab-Ezzouar, 16111, Algiers, ALGERIA, Fax:/Phone (+213) 21 24 78 92
dalila_info@yahoo.fr

[3]LSIS - UMR CNRS 6168
Domaine Universitaire de Saint-Jérôme, Avenue Escadrille Normandie-Niemen
13397 MARSEILLE CEDEX 20, France.

Abstract

In this work, a new scatter search-based approach is studied for the NP-Hard satisfiability problems, in particular for its optimization version namely max-SAT. The paper describes a scatter search algorithm enhanced with a tabu search component combined with a uniform crossover operator. The latter is used to identify promising search regions while tabu search performs an intensified search of solutions around these regions. Our objective is to achieve a good compromise between intensification and diversification in the search process. Empirical tests are performed on DIMACS benchmark instances. Numerical results show a considerable performance in favor of our new approach.

keywords: scatter search, tabu search, max-SAT, metaheurictics, hybridization.

1 Introduction

The Satisfiability problem is a core problem in both computational complexity theory and artificial intelligence discipline. Its wide application to the domain of artificial intelligence in automatic reasoning and other domains as VLSI and graph theory motivates the huge interest shown for this problem. Formally, given a collection of m clauses C= { C_1, C_2, ….. Cm} involving n Boolean variables x_1, x_2 ,…, x_n. The SAT problem is to determine whether or not there exists a truth assignment for C that satisfies the m clauses. A clause is a disjunction of literals. A literal is a variable or its negation. A formula in conjunctive normal form (CNF) is a conjunction of clauses. The formula is said to be satisfiable if there exists an assignment that satisfies all the clauses and unsatisfiable otherwise. In the latter situation, we are interested to other variants of SAT. We mention among them the maximum satisfiability problem (max-SAT) which consists in finding an assignment that satisfies the maximum number of clauses. The decision variants of both SAT and max-SAT are NP-Complete [Cook, 1971; Johnson, 1974].The interest focuses on the development and implementations of heuristics.

Many algorithms have been proposed and important progress has been achieved. These algorithms can be divided into two main classes:

- *Complete algorithms*: dedicated to solve the decision version of SAT problem. The well- known algorithms are based on the Davis-Putnam-Loveland procedure [Davis et al, 1962]. Satz [Li, 1997] is a famous example of a complete algorithm.
- *Incomplete algorithms*: they are mainly based on local search and evolutionary algorithms. GSAT [Selman et al, 1994], Tabu search [Mazure et al, 1997; Boughaci and Drias, 2005], simulated annealing (Hansen and Jaumard, 1990), genetic algorithms [Frank, 1994], GRASP [Pardalos et al, 1996], scatter search [Drias, 2001] and recently memetic algorithm [Boughaci et al, 2004] are examples of incomplete algorithms for SAT. These meta-heuristics are a good approach for finding a near solution of very large instances, in particular for unsatisfiable or unknown instances.

In this paper, a new scatter search variant is proposed for this problem. Its algorithmic backbone is a scatter search (SS) enhanced with a uniform crossover operator combined with a Tabu search (TS) improvement strategy. The uniform crossover operator is used to identify promising search regions while tabu search performs an intensified search of solutions around these regions. Our objective is to achieve a good compromise between intensification and diversification in the search process. Empirical tests are performed on DIMACS benchmark instances. Numerical results show a considerable performance in favor of our new approach.

2 The Scatter Search Variant

Scatter search [Laguna et al, 1999a, 1999b] is a population-based meta-heuristic like Genetic algorithms. It is an evolutionary method that constructs solutions by combining others**.** The approach starts with an initial population (collection of solutions) generated using both diversification and improvement strategies, then, a set of best solutions (reference set that incorporates both diverse and high quality solutions) are selected from the population. These collections of solutions are a basis for creating new solutions consisting of structured combinations of subsets of the current reference solutions.

Tabu search is one of the efficient methods for large combinatorial optimization problems. Given the search space, the method attempts to find a global minimum state. It is a general meta-heuristic that has been proposed by Fred Glover [Glover, 1986]. Like a local search, tabu search starts with an initial configuration generated randomly, then, the best neighbor solutions are selected. Tabu search uses also a list (called tabu list) to keep information about solutions recently selected, that, in order to escape the solutions already visited. In the case, in which, a tabu move applied to a current solution gives a better solution; we accept this move in spite of its tabu status by aspiration criterion. The search stops when the quality of the solution is not improved during a maximum number of iterations or when we reach the optimum global.

Our scatter search-based evolutionary approach starts with an initial population of individuals created using a diversification generator. Then, each individual makes tabu search to improve its fitness. After that, a set of individuals (reference set that incorporates both *good* and *bad* individuals) are selected from the current population. The resulting reference set has B1 high quality of individuals plus B2 low quality of individuals.

The reference set is a basis for creating new individuals consisting of uniform combination of subsets of the current reference individuals. We precise, that our subset generation method combines good individuals with bad ones (see example below). The uniform crossover operator is applied to all subsets of individuals of the current reference set.

After having built new combined individuals using the uniform crossover operator, the combined individual is returned to TS procedure to serve as an initial starting point which may be enhanced. With all this components: diversification generator, reference set selection, uniform crossover and intensified tabu search procedure, we hope to be able to achieve a good compromise between intensification and diversification in the search process. The search terminates after a certain number of generations or when we reach the optimum global.

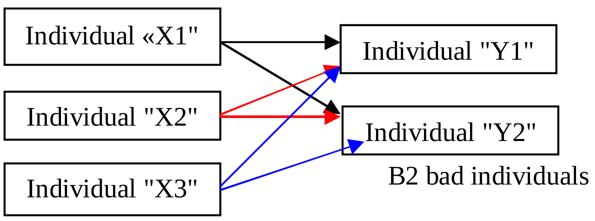
- A diversification generator: to generate diverse solutions, the generator creates, from a seed solution V, a collection of solutions associated with an integer h (1<h<=n).

Two types of solutions V' and V'' are created from V:

- Type1: V'[1+k*h]= 1- V[1+k*h], k=1,2,…..n/h, k<n.
- Type2 : V'' are the complement of V'.

- A subsets generation method: A subset generation method operates on the reference set to produce a subset of its individuals as basis for creating combined individual. In order to explore diverse regions, we propose to combine good individuals with bad ones.

Example:
Let consider a reference set consisting of three good individuals (X1, X2, X3) and two bad individuals (Y1, Y2).



B1 Good individuals

The subsets are generated as: {(X1,Y1), (X1,Y2), (X2,Y1), (X2,Y2), (X3,Y1), (X3, Y2)}

- A uniform crossover operator : As you know a crossover operator of a genetic algorithm selects a position K to be the crossing point from the range [1,…n]. The first K elements are copied from the one parent while the second part is copied from the second parent to create the new trial individual. Unlike this operator, our uniform crossover operator chooses bits from both parents without any crossover point. If the two bits have the same value, the child receives the common value bit. Otherwise, it receives a random value (0 or 1).

A Scatter Search Variant outline

The algorithm, depicted below, operates as follows. At the beginning of the search, our approach uses a diversification generator to build a population, P, of Psize individuals. The generator has focused on the diversification strategy and not on the solution quality. In step 4.1, a reference set Refset is created. A Refset is a collection of B individuals selected from P. The resulting reference set has B1 high quality individuals and B2 low quality of individuals (B=B1+B2). The uniform crossover is applied in step 4.3 to all subsets in the current reference set. The steps 4.3, 4.4 and 4.5 are performed as long as at least one new trial solution is admitted in the reference set. In the case in which, no new solution to be added to the reference set, step 4.6, rebuilds the reference set. The algorithm terminates after a certain number of iterations.

Step1. Initialization
Set scatter search variant parameters
//Psize= size of the population P,
//B=size of the reference set =B1+B2,
//B1= high quality subset selected according to their fitness function value, B2= bad subsets of individuals,
//maxiter= maximum number of generations,
//iter =current generation,
iter=1 ;
P= Φ ;
Step2. Diversification generator; Use **TS** to create enhanced trial individuals
Step3. Evaluate and order the individuals in P according to their objective function value
Step4. Iteration
While (iter < maxiter)
Begin
Step 4.1 Reference Set selections
Step 4.2 Subset Generation Method
While (continuing to maintain and update Refset) do
Begin
Step4.3. For each subset produced, use the uniform crossover operator to produce new individuals;
Step4.4 Use **TS** to create enhanced trial individuals
Step4.5. Update Reference Set
If the resulting individual improves the quality then add it to the B1 high quality individuals and remove the worst one. Else add it to the B2 individuals and removes the best one in B2.
end
Step4.6 Build a new population P by initializing the generation process with the reference set;
Use **TS** to create enhanced new trial individuals;
iter= iter+1;
End;
Step5. Termination, print the best solution with the best cost.

3. Computational Results

All experiments were run on a 350 Mhz Pentium II with 128 MO RAM. All instances have been taken from the SATLIB Web site. They are hard Benchmark Problems. On each instance the SSV algorithm has been executed in order to compute the average of the maximum number of satisfied clauses. The tables below show the results obtained by our algorithm. These columns contain the number of variables, the number of clauses, the number of satisfied clauses, the rate of satisfied clauses and the running time in second.

Two kinds of experimental tests have been undertaken. The goal of the first one is the setting of the different parameters of the SSV algorithm like the type of crossover, the number of iterations, the population size and the interaction between TS and SSV algorithms parameters. These parameters are fixed as: the maximum number of generations was set to maxiter=5, the population size was set to Psize = 50, the reference set was set to 10 (B1= 5, B2= 5) and the TS parameters was set as: maximum number of iteration = 30 and Tabu tenure=7.

The second kind of experiments concerns max-SAT instances. All these instances are CNF formula encoded in DIMACS CNF format (see, www.satisfiability.org)
The results found are classed by class

- **JNH class:** Randomly generated instances- constant density model. The instances have originally been contributed by John Hooker.

Table2. Solutions quality and running time results obtained by SSV on JNH instances.

Instances	# Var	#clauses	SSV-sol	Rate %	Time
Jnh201- yes	100	800	800	100	831,7
Jnh202- no	100	800	797	99	800,0
Jnh203-no	100	800	798	99	700,0
Jnh204-yes	100	800	798	99	799,3
Jnh205-yes	100	800	800	100	844,5
Jnh206-no	100	800	799	100	848,7
Jnh207-yes	100	800	799	99	790,4
Jnh208-no	100	800	798	99	225,7
Jnh209-yes	100	800	800	100	635,7
Jnh210-yes	100	800	800	100	218,3

- **AIM class**: Artificially generated random 3-SAT, defined by Kazuo Iwama, Eiji Miyano and Yuichi Asahiro. We have chosen four Satisfiable instances.

Table1. Solutions quality and running time results obtained by SSV on AIM instances.

Instances	# var	#clauses	SSVSol	Rate %	Time
Aim-50-1	50	80	80	100	20,0
Aim-50-2	50	100	99	99	34,1
Aim-50-3	50	170	170	100	0,7
Aim-50-6	50	300	300	100	1,1

Above, some results found by the scatter search variant Algorithm. The results obtained by the evolutionary approach are acceptable. In general, we can observe the superiority of the approach in solving max-SAT problems from the performance point of view. When intensified TS, and diversified components are incorporated in SS the solutions space is better searched. We precise, that in general, the role of a local search technique in scatter search is to locate the solution more efficiently.

4. Conclusion and Future Work

In this paper, we have presented the evolutionary meta-heuristic called Scatter Search. We have proposed to hybridize it with a tabu search improvement technique combined with a uniform crossover to solve the max-SAT optimization problems. Our objective is to achieve a good compromise between diversification and intensification in the search process. The solutions found by the hybrid method are encouraging; the result method takes its superiority from the use of several techniques as diversification used by scatter search components (generator, combination methods) and intensification strategy by the TS procedure.

References

[Boughaci and Drias, 2005], D.Boughaci H.Drias "Efficient and Experimental Meta-heuristics for MAX-SAT Problems". *LNCS, Volume 3503 / 2005*,, WEA 2005: pp:501-512.

[Boughaci et al, 2004]. Dalila Boughaci, Habiba Drias and Belaid Benhamou, "Solving Max-SAT Problems Using a memetic evolutionary Meta-heuristic", *in proceedings of 2004 IEEE CIS 2004, pp480-484,2004..*

[Cook, 1971] S. Cook. "The Complexity of Theorem Proving Procedures", *in the Proceedings of the 3rdAnnual ACM Symposium on the Theory of Computing, 1971.*

[Davis et al, 1962], M. Davis, G. Putnam, and D. Loveland "A machine program for theorem proving", *communication of the ACM*, 394-397, Jul 1962.

[Drias, 2001] Habiba Drias. "Scatter search with walk strategy for solving hard Max-W-Sat problems", *in proceedings of IEA- AIE2001, lectures note in computer science, LNAI-2070, Springer*, Budapest, 35-44, 2001.

[Frank, 1994] J. Frank, "A study of genetic algorithms to find approximate solutions to hard 3CNF problems", *in proceedings of Golden West international conference on artificial intelligence, 1994*.

[Johnson, 1974.]. S. Johnson. "Approximation algorithms for combinatorial problems", Journal of Computer and System Sciences 9, 256-278, 1974.

[Hansen and Jaumard., 1990] P.Hansen and B. Jaumard. "Algorithms for the Maximum Satisfiability", journal of computing 44-279-303, 1990.

[Glover, 1986]. Fred Glover. "Future paths for integer programming and links to Artificial intelligence", operation research vol 31. 1986.

[Laguna, Glover, 1999a] Manuel Laguna and Fred Glover, "*Scatter Search"*, Graduate school of business, University of Colorado, Boulder, 1999.

[Laguna et al 1999b]Manuel Laguna, Rafeal Marti.; V. Campos, "Scatter Search for the linear ordering problem", University of Colorado, Boulder.1999

[Li. and Anbulagan, 1997]. C. Li. and Anbulagan,"Heuristic based on unit propagation for satisfiability", *in Proceedings of CP 97, springer-Verlag, LNCS* 1330, pp 342-356, 1997.

[Mazure et al, 1997] B. Mazure, L. Sais, E. Greroire, " A Tabu search for Sat", *in proceedings of AAAI, 1997.*

[Pardalos et al, 1996] .PM. Pardalos. L.Pitsoulis, MGC. Resende "A Parallel GRASP for MAX-SAT Problems". *PARA96 Workshop on Applied Parallel Computing in Industrial Problems and Optimization, Denmark August 18-21, 1996.*

[Selman et al, 1994] Barti Selman, A. Henry, Z.Kautz, B. Cohen „Noise Strategies for Improving Local Search", *in Proceedings of AAAI*-94, 1994.