Improving the Asymptotic Convergence of Memetic Algorithms
The SAT Problem Case Study

Noureddine Bouhmala
Department of Maritime Technology and Innovation, Vestfold University College, P.O. Box 2243, 3103 Tnsberg, Norway

Keywords: Satisfiability Problem, Memetic Algorithm, Multilevel Techniques.

Abstract: In this work, a memetic algorithm that makes use of the multilevel paradigm for solving SAT problems is presented. The multilevel paradigm refers to the process of dividing large and difficult problems into smaller ones, which are hopefully much easier to solve, and then work backward towards the solution of the original problem, using a solution from a previous level as a starting solution at the next level. Results on real industrial instances are presented.

1 INTRODUCTION

The Satisfiability (SAT) problem which is known to be NP-complete (Cook, 1971) plays a central problem in many applications in the fields of Very Large Scale Integration (VLSI) Computer-Aided design, Computing Theory, and Artificial Intelligence. Generally, an instance of the SAT problem is defined by a set of Boolean variables $V = \{v_1, ..., v_n\}$ and a Boolean formula $\Phi : \{0,1\}^n \rightarrow \{0,1\}$. The formula Φ is in conjunctive mormal form (CNF) and is formed of a conjuction of clauses. Each clause in turn is a disjunction of literals and a literal is a variable or its negation. The task is to determine whether there exists an assignment of values to the variables under which Φ evaluates to True. Such an assignment, if it exists, is called a satisfying assignment for Φ, and Φ is called satisfiable. Otherwise, Φ is said to be unsatisfiable.

2 THE MULTILEVEL MEMETIC ALGORITHM (MMA)

2.1 General Strategy

The multilevel paradigm involves recursive coarsning to create a hierarchy of increasingly smaller and coarser versions of the original problem. This phase is repeated until the size of the smallest problem falls below a specified coarsening threshold. Then, a solution for the problem at the coarsest level is generated, and then successively projected back onto each of the intermediate levels in reverse order (uncoarsening phase). The solution at each child level is improved by a refinement algorithm before moving to the parent level.

2.2 Coarsening

Let S_0 (the subscript represents the level of problem scale) be the set of literals. The next level coarser level S_1 is constructed from S_0 by merging literals. The merging is computed using a randomized algorithm similar to (Hendrickson and Leland, 1995). The literals are visited in a random order. If a literal l_i has not been matched yet, then we randomly select one randomly unmatched literal l_j, and a new literal l_k (a cluster) consisting of the two literals l_i and l_j is created. Unmerged literals are simply copied to the next level. The new formed literals are used to define a new and smaller problem and recursively iterate the coarsening process until the size of the problem reaches some desired threshold.

2.3 Initial Solution

The coarsening procedure ceases when the problem size shrinks to a desired threshold. Initialization is then trivial and consists of generating an intial solution for the population at the lowest level $|level_m|$ using a random procedure. Each cluster of every individual in the population is assigned the value of true or false in a random manner.

Bouhmala N..
Improving the Asymptotic Convergence of Memetic Algorithms - The SAT Problem Case Study.
DOI: 10.5220/0004153702940296
In *Proceedings of the 4th International Joint Conference on Computational Intelligence* (ECTA-2012), pages 294-296
ISBN: 978-989-8565-33-4

2.4 Uncoarsening

The uncoarsening phase refers to the inverse process followed during the coarsening phase. Having improved the assignment at the level $level_{m+1}$, the assignment must be extended on its parent $level_m$. The extension algorithm is simple; if a given cluster C_i belonging to an individual in the population at the level j is assigned the value of true then the merged pair K_l and K_m of clusters that it represents, are also assigned the true value.

2.5 Improvement

The idea of improvement is to use the projected population at $level_{m+1}$ as the initial population for $level_m$ for further refinement using a memetic algorithm described in the next section. Even though the population at the $level_{m+1}$ is at a local minimum, the projected population may not be at a local optimum with respect to $Level_m$. The projected population is already a good solution and contains individuals with high fitness value, MA will converge quicker within a few generation to a better assignment. As soon as the population tends to loose its diversity, premature convergence occurs and all individuals in the population tend to be identtical with almost the same fitness value. During each level, the memetic algorithm is assumed to reach convergence when no further improvement of the best solution has not been made during five consecutive generations.

3 EXPERIMENTAL RESULTS

3.1 Test Suite

The performance of the multilevel memetic algorithm (MMA) was tested on a set of large problem instances taken from real industrial problems. All the benchmark instances used in this experiment are satisfiable instances. Due to the randomization nature of the algorithms, each problem instance was run 100 times. Time limit is set to 15 minutes. The tests were carried out on a DELL machine with 800 MHz CPU and 2 GB of memory. The code was written in C and compiled with the GNU C compiler version 4.6. The following parameters have been fixed experimentally and are listed below:

- Crossover probability = 0.85;
- Mutation probability = 0.1;
- Population size = 50;
- Stopping criteria for the coarsening phase: The coarsening stops as soon as the size of the coarsest problem reaches 100 variables (clusters). At this level, MA generates an initial population.
- Convergence during the refinement phase: If there is no observable improvement of the fitness function of the best individual during 10 consecutive generations, MA is assumed to have reached convergence and moves to a higher level.

3.2 Analysis of Results

Table 1 shows the range of all solved clauses (RAC), the mean solved clauses (MSC) and the range of solved clauses (RSC). As can be seen in Table 1 there is no overlap between the observed ranges for MA and MMA. Hence all observed runs of MMA is found to be better (closer to the solution) than the runs of MA. The actual domination of MMA versus MA is strengthened by the fact that a none of the 99% confidence intervals for the mean difference between MMA and MA contains the value 0. Finally, we can see that MMA have better asymptotic convergence (to around $0.39\% - 0.95\%$ in excess of the solution) as compared to MA which only achieve around $(10,05\% - 11,95\%)$. We noticed that for small problems MA dominates MMA during the start of the search, however as the time increases, MMA has a marginally better asymptotic convergence for small problems compared to MA while the convergence behavior becomes more distinctive for larger problems.

4 CONCLUSIONS

A new approach for addressing the satisfiability problem which combines the multilevel paradigm with a simple memetic algorithm have been tested. A set of industrial benchmark instances was used in order to get a comprehensive picture of the new algorithm's performance. The multilevel memetic algorithm clearly outperformed the simple memetic algorithm in all instances. Results also show that the difference in performance between the two algorithms increases for larger problems. The experiments have shown that MLMA works well with a random coarsening scheme combined with a simple memetic algorithm used as a refinement algorithm. The random coarsening provided a good global view of the problem, while the memetic algorithm used during the refinement phase provided a good local view. It can be seen from the results that the multilevel paradigm greatly improves the memetic algorithm and always returns a better solution for the equivalent runtime.

Table 1: Comparison of MMA and MA: Range of all solved clauses (RAC), Mean of solved clauses (MSC), Ranged of solved clauses (RSC).

		RAC		MSC		RSC	
	Instance	MA	MMA	MA	MMA	MA	MMA
1	2bitadd_10	0%	0%	98.85%	99.86%	98.24 – 99.23%	99.79 – 99.93%
2	2bitadd_11	0%	4%	98.96%	99.89%	98.46 – 99.49%	99.74 – 100%
3	2bitadd_12	8%	18%	99.91%	99.92%	99.76 – 100%	99.82 – 100%
4	2bitcomp_5	100%	100%	100%	100%	—	—
5	2bitmax_6	44%	56%	99.92%	99.94%	99.74 – 100%	99.74 – 100%
6	3bitadd_31	0%	0%	96.47%	99.58%	94.83 – 97.05%	99.31 – 99.61%
7	3bitadd_32	0%	0%	95.77%	99.58%	93.89 – 96.52%	99.42 – 99.62%
8	3blocks	0%	0%	99.93%	99.96%	99.91 – 99.96%	99.91 – 99.98%
9	4blocks	0%	0%	99.97%	99.98%	99.96 – 99.98%	99.97 – 100%
10	4blocksb	0%	0%	99.97%	99.98%	99.97 – 99.98%	99.97 – 99.99%
11	e0ddr2-10-by-5-1	0%	0%	89.53%	99.67%	88.09 – 90.47%	99.33 – 99.75%
12	e0ddr2-10-by-5-4	0%	0%	89.50%	99.69%	88.72 – 90.25%	99.31 – 99.74%
13	enddr2-10-by-5-1	0%	0%	89.20%	99.67%	88.46 – 90.08%	99.08 – 99.75%
14	enddr2-10-by-5-8	0%	0%	89.24%	99.68%	88.18 – 90.18%	99.15 – 99.76%
15	ewddr2-10-by-5-1	0%	0%	89.09%	99.66%	88.33 – 89.92%	99.05 – 99.76%
16	ewddr2-10-by-5-8	0%	0%	89.02%	99.71%	88.05 – 89.95%	99.42 – 99.76%

Our future work aims at investigating other coarsening schemes and identifying other parameters which may influence the interaction between the memetic algorithm and the multilevel paradigm.

REFERENCES

Cook, S. A. (1971). The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, STOC '71, pages 151–158, New York, NY, USA. ACM.

Hendrickson, B. and Leland, R. (1995). A multilevel algorithm for partitioning graphs. In *Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM)*, Supercomputing '95, New York, NY, USA. ACM.