**Title:** Improving Linear Search Algorithms with Model-based Approaches for MaxSAT Solving
**Authors:** R. Martins, V. Manquinho and I. Lynce
**Affiliation:** IST/INESC-ID, University of Lisbon, Portugal
**Journal:** Journal of Experimental & Theoretical Artificial Intelligence
**Responsible Editors:** Toni Mancini, Marco Maratea, and Francesco Ricca

# Reply to the Reviewers Comments

Dear Guest Editors and Reviewers,

We attach below our reply to the reviewer's comments to the RCRA 2013 JETAI version of our paper. We have introduced modifications in the paper resulting from comments made by the reviewers. We thank the reviewers for those comments.

The main updates from the previous version are the following: (i) overall revision of the text and clarification of the issues mentioned in the reviews, (ii) revision of the proofs and an additional proof for Algorithm 4, (iii) revision of the result analysis section with a clear split between instances from the MaxSAT evaluation and other instances, (iv) results for MAXHS MaxSAT solver were added and results for BCD2 were updated, and (v) references were updated as requested. Next, we detail the reply to all reviewer's comments.

## Reviewer #1

> *" The authors propose and analyze some new methods for partitioning the relaxation variables added to soft clauses for solving the MaxSAT problem. The paper is well-written and the results reported deserve to be published. The reasons of the difference of performances among the various methods are clearly discussed. The material and presentation is improved wrt the RCRA version. "*

We thank the reviewer for the comments.

> *" Some Lemma is really trivial and can be omitted. "*

The Lemmas and proofs have been modified.

> *" I have a general question on the applicability of a bisection scheme instead than a linear increasing one. Perhaps the authors can briefly comment on that. "*

Binary search approaches have been applied to linear search algorithms with some success. See for example:

X. An, M. Koshimura, H. Fujita, and R. Hasegawa. QMaxSAT version 0.3 & 0.4. In Workshop on First-Order Theorem Proving, pages 715. Bern, Switzerland, 2011.

A model-guided approach for binary search algorithms could also be devised. Independently and after the submission of this paper, model-guided approaches have been studied in the context of binary algorithms:

A. Morgado, F. Heras, and J. Marques-Siva. Model-Guided Approaches for MaxSAT Solving. In International Conference on Tools with Artificial Intelligence, November 2013.

They report improvements when using model-guided approaches for binary algorithms, such as BCD2. Model-guided approaches for this solver increased its performance for the close_solution benchmarks while deteriorating its performance for the ms_industrial benchmarks. However, the authors are only using a basic model-guided approach. As future work, we propose to study the effectiveness of an `Hybrid[All]` approach when applied to binary search algorithms.

*" Minor comments*

*Page 2. After "solution or model" Add the notion of satisfiable/unsatisfiable formula.*

*"can have weight" maybe "can have integer weights"*

*"multiset of clauses" I would add "(i.e., we allow multiple occurrence of the same clause in a formula"*

*Page 3. Def. 2.3 Remove "(i.e., ... formula)" - see comment above.*

*"the sum of n literals" Please clarify here that, i.e., true + true = 2*

*"We refer to the literature for B&B [1] and unsatisfiability based [2] algorithms." → "We refer to [1] for B&B algorithms and to [2] for unsatisfiability based [2] algorithms, respectively". "*

The above suggestions have been considered.

*" Page 4. Algo 1, Output. "satisfying assignment" it is not defined. Rather, you have defined it as a solution or model. "*

Following the suggestion of the reviewer, we have replaced "satisfying assignment" by "model".

*" Why are you using "st" as variable name? "state"? Can you add a (minor of course) comment?"*

"st" denotes the status of the solver (satisfiable or unsatisfiable). This is now made clear in the paper.

*" Page 6. Algo 2. You can remove $V_R$ from instruction 1. It is initialized in instruction 2.*

*Page 10. Algo 3. Same observation above on $V_R$. "*

The `relaxFormula` procedure adds variables to $V_R$. However, before giving $V_R$ to the `relaxFormula` procedure, we should initialize $V_R$ to empty.

*" Page 7. The intensional set $\omega_i \in \phi_s : \omega_i \cup r_i$ should be: $\omega_i \cup r_i : \omega_i \in \phi_s$ Btw, this is the unique place where you use ":" (in the other sets you're using "|").*

*References. Missing editor in Lin and Su, missing LNCS Vol ... in Sinz.*

*Typos*

*Page 1. "bioinformatic" → "Bioinformatics"*

*Page 5. "the ones" maybe "those"*

*Page 9. "As with previous" → "As previous"*

*Page 13. "active ." → "active."*

*Page 18. "variables [all] corresponds" → "variables, respectively, [all] corresponds"*

*"*

The above suggestions have been taken into consideration.

## Reviewer #2

*" I recommend accepting the manuscript for publication subject to a major revision, taking into account all of the following comments.*

*The most critical issue needing revision:*

*I am not 100% convinced about the claim that the (best-performing variant of) the algorithms proposed improves on the state-of-the-art in MaxSAT solving. This is due to two things:*

*(1) Austin appears to improve the best current MaxSAT solvers (esp. MSU-BIN-C-D) only on a *single* problem domain, namely, "close solutions". The authors should provide arguments why this is especially interesting, or better, include more problem domain and provide more fine-grained views to the results on the individual MaxSAT Evaluation problem domains. A more "standard" way would be to evaluate the algorithms only on the MaxSAT Evaluation instances, in which case Austin would not improve over the state of the art. The more benchmarks, the better, of course, but the benchmark selection should be somewhat balanced: Now there is a clear imbalance in the benchmark selection in view of Figure 3, since the number of problem instances from the "close solutions " domain is high compared to the *total* number of miscellaneous MaxSAT Evaluation instances. Hence the claim "The results are clear: Austin outperforms all other MaxSAT solvers ..." is quite misleading. "*

The main goal of our paper is to improve the performance of linear search algorithms for MaxSAT instances with a large number of soft clauses.

Our goal is not to improve the performance of linear search algorithms when the number of clauses is small, but simply to maintain their performance on these instances. The benchmarks from pms_industrial benchmarks from the MaxSAT evaluation have a small average number of soft clauses (see Table 2). Therefore, our goal is not improve the performance of linear search algorithms on this benchmark set.

We have shown that the proposed algorithms improve linear search algorithms on MaxSAT instances with a large number of soft clauses. For example, the best-performing algorithm can solve 89 ms_industrial benchmarks and 262 close_solution benchmarks, whereas the linear search algorithm can only solve 48 ms_industrial benchmarks and 196 close_solution benchmarks. At the same time the best-performing algorithm have a similar performance on the pms_industrial benchmarks, where it only solve 1 less instance than the linear search algorithm.

Moreover, the best-performing algorithm is competitive with state-of-the- art MaxSAT solvers, and can improve their performance when the number of soft clauses is large.

We have introduced modifications throughout the paper to clarify the goal of the proposed algorithms.

*" Is "close solutions" more interesting as a problem domain than the individual problem domains constituting the MaxSAT Evaluation instance sets? How does Austin compare to the other solvers on the individual problem domains? "*

The motivation behind model-based algorithms is to solve MaxSAT instances with a large number of soft clauses. The pms_industrial benchmark set consists of instances that have a small average number of soft clauses (see Table 2).

Therefore, it is not expected that model-based algorithms improve the performance of linear search algorithms on the pms_industrial benchmark set. The close_solution benchmarks have the property of having a large number of soft clauses, which is the property that we want to test our algorithms on.

*" Is MaxSAT the best approach to solving "close solutions"?"*

Our goal is not to improve other approaches that can be used to solve the close_solution benchmarks, but to improve classic linear search algorithms when the number of soft clauses is large.

Ignasi Abío compared the performance of MaxSAT 2011 solvers with a conflict directed lazy decomposition approach which is implemented on top of the SMT solver Barcelogic. At that time, their approach outperformed MaxSAT solvers for the close solution benchmarks.

We did not compare AUSTIN with their approach but looking at the results provided in their paper, it seems AUSTIN would be competitive with the approach implemented on Barcelogic.

*" How is the problem encoded in MaxSAT? "*

The close solution problem is encoded in MaxSAT by taking the original solution and encoding each literal as a unit soft clauses. New clauses are added as hard clauses. The goal is to find a new solution that is as close as possible as the original solution.

*“ What are the underlying SAT instances from which the “close solutions” MaxSAT instances have been based on, and how was the “first solution” obtained? ”*

> The close solution benchmarks were provided by Ignasi Abío and more details can be found in the following paper:
>
> I. Abío and P. J. Stuckey. Conflict Directed Lazy Decomposition. In Principles and Practice of Constraint Programming, pages 70−85, 2012.
>
> The underlying SAT instances come from timetabling and from satisfiable instances from the industrial category of the SAT competition 2011. For the latter, the original solution is found by a SAT solver. A new clause is then generated that falsifies the original solution.

*“ (2) The evaluation is critically missing the recently proposed MaxSAT solver MaxHS, http://www.cs.toronto.edu/∼jdavies/daviesCP13.pdf (and earlier papers at SAT’13, CP’11). This is based on a different type of a hybrid algorithm, and I look forward to seeing how it performs in comparison on each of the benchmark sets. This evaluation needs to be added. ”*

> MAXHS is now included in the evaluation. This solver performed well for the pms_industrial benchmark set, where it solves less two instances than BCD2.
>
> On the other hand, it performed poorly on instances with a large number of soft clauses, namely, on the ms_industrial benchmarks, and on the close_solution benchmarks.

*“ More detailed comments in chronological order:*

*Section 1:*

*-“A preliminary version of this work was published ... . In previous work”. Please revise these sentences. It is unclear to me to what “In previous work” refers to. Was the RCRA paper formally published somewhere?*

*-The introduction should be extended with a thorough discussion on “previous work” with references, so as to make the sentence “This paper extends previous work ...” more explicit. ”*

> Previous work was referring to the RCRA 2013 workshop paper. This work has not been formally published anywhere else. This section has been revised to clarify this issue.

*“ Section 2:*

*-“... as being a multiset of clauses”. This is a strange sentence. My guess is that you are trying to say that (integer) weighted MaxSAT can be viewed like this by duplicating clauses according to the weights? ”*

> We wanted to denote that a clause may appear more than once. This sentence has been changed to clarify this issue.

*“ -Instead of $\varphi = \varphi_h \cup \varphi_s$, why not view $\varphi$ as a pair $(\varphi_h, \varphi_s)$ ? ”*

Typically, a MaxSAT formula is viewed as one formula instead of a pair of formulas. This is mainly due to the input format accepted by MaxSAT solvers.

*“ -Definitions 2.1-2.3 are repeating to an extent what was already defined. Please compact. ”*

The definitions were replaced by a more compact text.

*“ -“An optimal solution ... would be”. would be $\rightarrow$ is.*

*-$\sum_i = 1^k x_i$. Replace $x_i$ with $l_i$, and explicate that *$l_i$* is a literal. ”*

The above suggestions have been taken into consideration.

*“ -“unsatisfied” is not defined. Should be “not satisfied”. (repeatedly in the paper) ”*

An unsatisfied clause is defined on page 2: “A clause is said to be unsatisfied if all of its literals are unsatisfied.”

*“ -”Although cardinality constraints do not occur in MaxSAT formulations” is a strange sentence. Definitely there are underlying cardinality constraints is various problem that have been encoded as MaxSAT. Please revise. ”*

The above sentence has been revised.

*“ Section 2.1:*

*-include MaxHS as a reference. ”*

MAXHS is now included in the MaxSAT algorithms section.

*“ - “Since this paper focus*es* on”. Please provide justifications for *why* this is the focus. Are they competitive? ”*

Linear search algorithms are very competitive for solving industrial partial MaxSAT benchmarks. For example, QMAXSAT (a known MaxSAT solver that uses a classic linear search algorithm) have won the industrial category of the partial MaxSAT Evaluation for several years.

*“ -“linear search *over*” (not “on”)*

*-“Algorithm 1 shows” → “...is presented as Algorithm 1”.*

*-“problem variables” not defined.*

*-“no satisfiable call” revise. This base case should be explain as one of the first things.*

*-Example 2.5 (and the later examples): the sentence flip back of forth between the active “we” and the passive. Please revise by removing “we”. It is the algorithm, not we. ”*

The above suggestions have been considered.

*“ Section 3:*

*- “in contrast *to*” ”*

Done.

*“ 2nd sentence: “In contrast with traditional linear search algorithms, model-based algorithms use models given by the SAT solver to iteratively increase the set of relaxation variables that are used in the cardinality constraint.” One sentence after that: “The first algorithm uses the models given by the SAT solver to iteratively increase the set of relaxation variables that are used in the cardinality constraint.” Please revise. Right now it looks like the first algorithm is exactly what “model-based algorithms” are as you define them. ”*

This paragraph has been revised.

*“ -“a partial partition *of*”*

*-“Even t*h*ough” ”*

Done.

*“ -Section 3.1. title “Model-based Algorithms” is repetitive. ”*

The section titles have been revised.

*“ -“Algorithm 2 shows our adaptation” what did you adapt, and how? ”*

Both MinSAT and MaxSAT can be solved by linear search algorithms. The main difference relies on the relaxation of the soft clauses.
We adapted the model-based algorithm from MinSAT to MaxSAT by changing the relaxation procedure to the one used in linear search algorithms for MaxSAT.

*“ -For Defs 3.1-3.2, I suggest defining a “stage” of the algorithm, and the referring to “at any stage”, so that the definitions are more explicitly tied with the execution. ”*

A “stage” of the algorithm has been defined and the definitions changed accordingly.

*“ “the algorithm” should be “Algorithm 1”.*

*“did not found” $\rightarrow$ at this stage the SAT solver calls have not produced a model that assigns r to 1?*

*“did not found” would be “did not find” or better “has not yet returned”. ”*

The above suggestions have been taken into consideration.

*“ -“Alg 2 may require additional SAT calls” how many in the worst case? What happens in practice? ”*

In the worst case, the number of additional SAT calls is linear in the number of soft clauses. In practice, the number of additional SAT calls is much smaller.

*“ -“Proof sketch” is not ok for a full-length journal article. please revise. ”*

The proofs have been revised.

*“ -“As a result, we have ...” the following set definition seems to be “the wrong way”, what is after “:” should be before, and vice versa? ”*

This set definition has now been fixed.

*“ -Lemma 3.4: should be “$\varphi_R$ is satisfiable iff $\varphi_h$ is satisfiable”.*

*-“can easily define” remove “easily”. ”*

The above suggestions have been taken into consideration.

*“ -Lemma 3.5: “The converse” is too implicit and informal. Please be more specific. split the lemma into two parts. ”*

The above suggestion was considered and the Lemma was split into two parts.

*“ -“is minimum” over what? please revise. ”*

Is minimum with respect to the total number of relaxation variables. The sentence has been revised to clarify this issue.

*“ -Lemma 3.6: here you will also benefit from defining a “stage” of the algorithm. However, you want to say that after a rel. variable becomes active, it never becomes inactive (will remain active).*

*-Lemma 3.6. should be explicitly connected to Alg 2. ”*

The Lemma was revised.

*“ -Lemma 3.7: “the algorithm” → Algorithm 2.*

*-Thm 3.8: remove “model-based”. “*the* correct solution” is incorrect. Should be *a*. ”*

Done.

*“ Section 3.2:*

*-“latter use” → later use.*

*-Algorithm 3, line 22: ∩ should be ∧? Same problem in alg 4, line 25.*

*-“the these” remove “the”. ”*

The above suggestions have been considered.

*“ -Lemma 3.13: “improved on” not defined? ”*

A definition was added in section 2.

*“ -Lemma 3.15: “Otherwise, ...” sentence appears redundant here? In any case, revise. ”*

It was not redundant. Nevertheless, the lemma was revised for clarity.

*“ -Thm 3.17: remove “model-based”. ”*

Done.

*“ Section 3.3:*

*-“active .” remove space (two occurrences)*

*-“we are reducing” is too definitive. Please revise. Also, should be “we reduce”.*

*-“truth value 1,” remove “,”. ”*

The above suggestions have been considered.

*“ -Theorem on the correctness of Algorithm 4 is missing, and must be provided. ”*

The correctness of Algorithm 4 is now included.

*“ Section 4:*

*-Def 4.1 is extremely vague, and not really a definition at all. “the dependencies between assignments” should be revised. Either crisply define what you want to define, or use the somewhat standard terminology along with appropriate references and pointers. ”*

Definition 4.1 has been removed. We now use the standard terminology and point to appropriate references.

*“ -The paragraph on (non-)incremental SAT solving is vague and badly worded. Please revise. “is denoted by” should revised to “we call” or similar. (other occurrences too) You should explicate that the underlying idea is to keep learned clauses and *still maintain correctness*. ”*

As suggested this paragraph has been revised.

*“ Section 5:*

*-“in a selection” please revise for more clarity. ”*

The benchmarks are described on section 5.1. This sentence has been revised to reflect that.

*“ -How robust/dependent is Austin wrt the choice of the SAT solver and cardinality encoding? Please add explicit data or at least discussion. Why did you make these choices, and based on what? ”*

We have selected Glucose 2.3 since it is a state-of-the-art SAT solver, and we have used cardinality networks since it is an encoding that requires less auxiliary variables/clauses than other cardinality encodings.

We did not evaluate AUSTIN with different SAT solvers or using different cardinality encodings. The goal of this paper is to evaluate the performance of model-based algorithms when compared to classic linear search algorithms. All algorithms have been implemented on top of the same baseline code. Therefore, the performance gains are due to the algorithms and not due to the SAT solver, or the cardinality encoding.

*“ -Please include explicit details on the close solution (see the main comments above for more). ”*

We have included more details on the nature of the close solution benchmarks.

*“ -”the problem that was observed” what does this refer to? please revise. ”*

The incremental and non-incremental versions of the Hybrid approach both rebuild the cardinality constraint more than once. Whereas, in the linear search algorithm, the incremental approach builds the cardinality encodings only once and the non-incremental approach rebuilds the cardinality encoding at each call to the SAT solver.

This paragraph have been revised to clarify this issue.

*“ -Section 5.4, solver listing: The various solvers used both different SAT solvers and different cardinality constraints. Has the effects of these choices been evaluated somewhere? Please provide appropriate references. Also, include explicit discussion pointing out these differences. Based on the relatively small differences in the performance of the solvers, I would like to know how large part of the differences is due to the choice of the SAT solver/encoding. ”*

To the best of our knowledge there is no evaluation of the impact of different SAT solvers and different encodings for each of the MaxSAT solvers. Each research group performs an internal evaluation and chooses the solver/encoding that is more suitable for their tool. Moreover, most of the MaxSAT solvers are not open-source and therefore it is not possible to perform such evaluation.

Note that, each solver is using a different search algorithm and the differences in performance are expected to be mostly due to the different algorithm and not to the different solver/encoding. We have added the Virtual Best Solver (VBS) to the cactus plot in Figure 4 which shows that the number of instances solved by all solvers is much larger than any individual solver.

*“ -”the most robust solver this outperforming” I don’t see this causal relation. Please revise. As already mentioned, as far as I can see, Austin is better than MSU-BIN-C-D only in a single problem domain.*

*-”The results are clear”. Not really. Please revise. ”*

This section have been revised to state the following:

AUSTIN significantly improves the performance of classic linear search algorithms for instances where the number of soft clauses is large, while maintaining their performance when the number of soft clauses is small.

Moreover, AUSTIN is competitive with state-of-the-art MaxSAT solvers, being particularly effective for the close_solution benchmark set.

*“ Section 6:*

*-”Linear search algorithms for MaxSAT have shown to be particularly effective when the number of soft clauses is small. However, if the number of soft clauses increases, then the performance of linear search algorithms tends to deteriorate.” I would have liked to have seen more detailed experimental data supporting this claim. Can you show (based on the actual data that you already have) an actual correlation between the number of soft clauses and the performance, as a plot? ”*

We have added scatter plots to section 5.2 that show the impact of the number of soft clauses in the performance of the algorithms. We have compared the classic linear search algorithm against our best performing model-based algorithm on instances with a small number of clauses and on instances with a large number of soft clauses. Both algorithms have similar performance on instances with a small number of soft clauses. For instances with a large number of soft clauses the model-based algorithm clearly outperforms the classic linear search algorithm.

*“ -“outperforms state-of-the-art”. This should be toned down, and revised to better reflect the results, including a comparison with MaxHS. ”*

MAXHS is now included in the evaluation and we have revised the above sentence to better reflect the results.

## Reviewer #3

*“ I have no special comments in fact. The authors really made an important effort to extend the approaches presented in the original RCRA paper (with the hybrid approach) and improve the experimental evaluation (larger benchmark, comparison with more state-of-the-art MaxSAT solvers). The extensions are clear, well described and significant. All points in the article are well justified. I particularly appreciate the illustration of the different algorithms with the ongoing example. I think this will be a very interesting article for the MaxSAT community. ”*

We would like to thank the reviewer for the kind words and appreciation of the paper.

## Reviewer #4

*“ The authors present different methods for advancing MaxSAT solving when tackled with linear search algorithms. At a first look the authors have added a good amount of new material in comparison to the workshop paper. In particular, they now analyze how hybrid approaches influence the overall behaviour of the solver, what is the role of incremental solving, and showed proofs of correctness and termination ”*

We thank the reviewer for the comments.

*“ However there are some issues in how the work is presented. First, I find a bit confusing how and when the authors talk about MaxSAT and Partial MaxSAT, it is not always clear what a technique is meant for. This affects abstract, introduction, and in smaller amount the preliminaries. ”*

The proposed algorithms target partial MaxSAT instances.

Note that MaxSAT is a particular case of partial MaxSAT. For simplicity, when we mention MaxSAT we are referring to a partial MaxSAT formula without any hard clauses. This is now clarified in the paper.

*“ I find the examples presented throughout the paper quite good, but their description is definitely lengthy, could be made much more concise. ”*

The examples have been revised and have been made more concise.

*“ Same partially applies to the experimental analysis section, where it would be nice to see a comparison of the uniquely solved problems. ”*

The number of uniquely solved problems is usually small. From the tested solvers, the MAXHS is the one that solves more uniquely solved problems since it is based on a hybrid approach between a MIP solver and a SAT solver.

We have added a virtual best solver (VBS) to the cactus plot. Even though the uniquely solved problems is small, the VBS shows that the number of instances solved by all solvers is much larger than any individual solver.

*“ What would you suggest to improve in model based approaches to make them stronger for pms_industrial problems? ”*

The motivation behind the model-based algorithms is to solve instances with a large number of soft clauses. The benchmarks in pms_industrial have a small number of soft clauses (see Table 2). Therefore, it is not expected that model-based approaches will improve the performance of classic linear search algorithms for these kind of benchmarks.

*“ Minor remarks:*

*footnote 1, ‘a’ in “all” is the only italic letter*

*P16: “we must rebuilt” → rebuild ”*

The above suggestions have been taken into consideration.

Lisbon, February 2014