Schemata Bandits for MAXSAT:

Conference Paper · July 2016

DOI: 10.1145/2908961.2931644

4 authors, including:

Nixon Ronoh
Vrije Universiteit Brussel

Edna Milgo
Moi University

Ambrose Kiprop
Moi University

Schemata Bandits for *MAXSAT* *

Nixon K. Ronoh †
Vrije Universiteit Brussel
Dept. of Comp. Sci.
Pleinlaan 2, 1050 Brussel
nronoh@vub.ac.be

Edna Milgo ‡
Vrije Universiteit Brussel
Dept. of Comp. Sciences
Pleinlaan 2, 1050 Brussel
emilgo@vub.ac.be

Ambrose K. Kiprop §
Moi University
Sch. of Bio and Phy. Sciences
P.O. Box 3900 , Eldoret, Kenya
ambkiprop@gmail.com

Bernard Manderick ¶
Vrije Universiteit Brussel
Dept. of Comp. Sciences
Pleinlaan 2, 1050 Brussel
bmanderi@vub.ac.be

ABSTRACT

In this paper, we propose the use of schemata bandits for optimization. This technique is a subclass of hierarchical bandits where the bandits are schemata. We investigate its use on a benchmark of binary combinatorial optimization problems, the Maximum Satisfiability *(MAXSAT)* problem. We compare performance with hierarchical Bayesian Optimization Algorithms *(hBOAs)* namely *GSAT* and *WALK-SAT*. Results suggest that using a bandit strategy enhances solver performance.

CCS Concepts

•**Computing methodologies** → **Sequential decision making;** •**Mathematics of computing** → *Optimization with randomized search heuristics;*

Keywords

Random search; Schemata bandits; MAXSAT; *hBOA*s

1. INTRODUCTION

1.1 The K-Armed Bandit

The Multi Armed Bandit *(MAB)* is an experiment with an aim of accumulating rewards from a pay-off distribution with unknown parameters, characterized by a reward function of a finite set of actions that are to be learned sequentially. At each stage of the stochastic process, a decision must be made on which of K arms to observe next [5, 6]. The experimenter decides between **exploiting** an arm that seems to be optimal and **exploring** whether other optimal arms exist, hence the *exploitation/exploration trade-off*. Action selection policies are used to balance this trade-off, with the aim of minimizing regret (the cost of playing an arm in place of the optimal).

1.2 The Monte Carlo Tree Search

Monte Carlo Tree Search ($MCTS$) seeks to optimize a problem domain by sampling from a decision space and constructing a search tree based on the results. Decision networks are constructed using a tree policy that provides a balance between exploration of nodes that have not been sampled and exploitation of nodes which appear to be optimal. Using previous exploration values, $MCTS$ progressively builds other partial trees then provides estimates of alternate actions, with the accuracy increasing as the tree is built. This continues until the nodes can be expanded no more. The optimal action is then returned [2, 1].

2. ALGORITHMS

2.1 GAs

In genetic algorithms *(GAs)*, a population is randomly generated from a candidate group of individuals. An evaluation is made based on fitness to select two **parents**, who reproduce by a *crossover* process to give an **offspring**. Random *mutation* ensures diversity and keeps duplication at minimal. This process continues until an optimum solution has been achieved, or until a given generation depth has been attained.

2.2 hBOAs

In hierarchical Bayesian Optimization Algorithms ($hBOAs$), the deterministic hill climber *(DHC)* evaluates the literal that improves the current solution until no further change is possible, a process known as $GSAT$. $WalkSAT$ is a technique that enhances $GSAT$ by incorporating random changes. Both GAs and $hBOAs$ are based on selection and recombination, but GAs could not as efficiently solve $MAXSAT$ problems.

*A full version of this paper is available at `www.acm.org/nronoh.htm/`

†Ronoh is a Ph.D student at VUB and lectures at Moi Uni.

‡Milgo is a Ph.D student at VUB and lectures at Moi Uni.

§Kiprop is Professor at SBPS, Moi University, Kenya

¶Prof.Dr. Manderick is a Profesor at The VUB, and is the Ph.D students' Promoter

GECCO'16 Companion July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.
DOI: http://dx.doi.org/10.1145/2908961.2931644

Table 1: Percent Satisfaction for 50 trials

App.	Q=1.6	Q=2.67	Q=3.01	Q=4.36	Q=4.41
sch.b	0.93750	0.90625	0.90365	0.92661	0.91383
rnd.s	0.93750	0.93125	0.90365	0.91284	0.90249
bdt.s	0.95000	0.93125	0.84053	0.94954	0.85488
GST	0.93750	0.91875	0.91694	0.92202	0.90930
WST	0.95000	0.91875	0.91694	0.92202	0.90476

2.3 Schema Bandits

Schema Bandits may be considered as hierarchical bandits whose arms are schemas. The focus is an $\ell-$ dimensional hypercube search space B^{ℓ} where ℓ is the length of the bit-string and $B = \{0, 1\}$. The search space $H \in (0, 1, *)$ is a hypercube subspace of B, with $*$ as a *'don't care'* symbol. The procedure entails: *Selection*, where a schemata selection policy is recursively applied until the most promising non terminal node is arrived at; *Expansion*, where one or more child nodes are added: *Simulation* from the new node using a default policy; and *Back-propagation* of the outcome through the selected nodes to update their statistics. The most general schema $[* * *...*]$ is the root, and specificity increases with the number of $*s$ replaced [3].

3. THE MAXSAT PROBLEM

MAXSAT seeks the total number of clauses that can be simultaneously satisfied from the set of all satisfiable Conjunctive Normal Form *(CNF)* expressions. Approaches that have been applied before include the satisfiability based approach, which decomposes each *MAXSAT* into multiple *SAT* problems and uses a *SAT* solver on each to obtain the actual solution. *BnB* utilizes a depth first branch and bound search in the space of possible assignments and uses an evaluation function at each search node to determine if there is a pruning opportunity [4]. The task is to determine whether a formula has a satisfying truth assignment.

4. EXPERIMENTAL COMPARISON

We considered $3-CNF$ benchmark instances from *SATLIB* library, performed experiments for the Schemata search algorithm *sch.s* and compared the results with those of a random search strategy *rnd.s*, a UCB1 based algorithm *bdt.s*, and hierarchical Bayesian Optimization Algorithms *GSAT* and *WALKSAT*. We utilized two main performance measures, namely *Regret* given the most optimum strategy, and the *Number of trials* before the optimum is attained.

5. RESULTS

The results are classified under Q, the ratio of the *number of clauses* to the *number of variables*. We determined the number of trials made before the optimum as shown in Table 1 for a budget of N trials. For $N = 50$, the most promising result for *sch.s* was at $Q = 1.6$, where a single trial was needed to find the optimum. At $Q = 4.41$, the most optimum value was observed with *sch.s* . The relative performance for $Q = 2.67$ is shown in Figure 1 , where *sch.s* is observed to have the least cumulative regret. The same performance was observed for $Q = 3.01$. The bandit based strategy gave the most optimum value at $Q = 1.6$, but with 44 trials of the budgeted $N = 50$ trials.

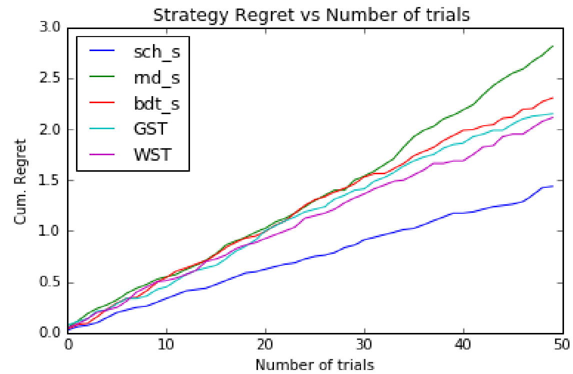

Figure 1: Algorithm comparison for 50 trials

6. CONCLUSIONS

Preliminary tests on benchmark *MAXSAT* instances show that the *Schemata bandits* technique indeed performs competitively as compared with the *hBOAs GSAT* and *WALKSAT*. The performance of the bandit based algorithm *bdt.s* puts into perspective the fact that hybridizing solvers for binary combinatorial problems with MAB strategies improves their performance.

7. ACKNOWLEDGMENTS

The authors would like to thank *VLIR-UOS*, *The Belgian Development Cooperation*, *Moi University* and *Vrije Universiteit Brussel* with whose sponsorship this study was made possible.

8. REFERENCES

[1] G. Chaslot, J.-T. Saito, B. Bouzy, J. Uiterwijk, and H. J. V. D. Herik. Monte-carlo strategies for computer go. *Proceedings of the 18th BeNeLux Conference on Artificial Intelligence, Namur, Belgium*, pp. 83-91, 2006.

[2] R. Coulom, P. Ciancarini, and H. J. van den Herik. Efficient selectivity and backup operators in monte-carlo tree search. *5th International Conference on Computer and Games*, inria-00116992: May 2006, Turin, Italy., 2006.

[3] M. M. Drugan, P. Isasi, and B. Manderick. *Simulated Evolution and Learning: 10th International Conference, SEAL 2014, Dunedin, New Zealand, December 15-18, 2014. Proceedings*, chapter Schemata Bandits for Binary Encoded Combinatorial Optimisation Problems, pages 299–310. Springer International Publishing, Cham, 2014.

[4] M. C. A. C. K. Pipatsrisawat, A. Palyan and A. Darwiche. Solving weighted max-sat problems in a reduced search space: A performance analysis. *Journal on Satisfiability, Boolean Modeling and Computation*, 4:191–217, 2008.

[5] S. L. Scott. A modern bayesian look at the multi armed bandit. *Appl. Stochastic Models Bus. Ind*, 26:639-658, 2010.

[6] J. Vermorel and M. Mohri. Multi armed bandit algorithms and empirical evaluation. *In European Conference on Machine Learning, Springer*, PP. 437-448, 2005.