Centre for Data Analytics

Insight

Acquiring Local Preferences of Weighted Partial MaxSAT

Hong Huang, Laura Climent and Barry O'Sullivan
Insight Centre for Data Analytics, University College Cork, Ireland

Abstract

Many real-life problems can be formulated as boolean satisfiability (SAT). In addition, in many of these problems, there are some hard clauses that must be satisfied but also some other soft clauses that can remain unsatisfied at some cost. These problems are referred to as Weighted Partial Maximum Satisfiability (WPMS). For solving them, the challenge is to find a solution that minimizes the total sum of costs of the unsatisfied clauses. Configuration problems are real-life examples of these, which involve customizing products according to a user's specific requirements. In the literature there exist many efficient techniques for finding solutions having minimum total cost. However, less attention has been paid to the fact that in many real-life problems the associated weights for soft clauses can be unknown. An example of such situations is when users cannot provide local preferences but instead express global preferences over complete assignments. In these cases, the acquisition of preferences can be the key for finding the best solution. In this paper, we propose a method to formalize the acquisition of local preferences. The process involves solving the associated system of linear equations for a set of complete assignments and their costs. Furthermore, we formalize the characteristics and size of the complete assignments required to acquire all local weights. We present an heuristic algorithm that searches for such assignments which performs promisingly on many benchmarks from the literature.

Background

The challenge of solving many real-life problems is often beyond simply finding a satisfiable assignment due to the preferences specified within the problem. Frequently, users specify preferences for certain combinations and the objective becomes finding the optimal solution. This solution maximizes the satisfiability of the specified preferences. Some typical examples of real-life applications that present such characteristics are configuration problems, scheduling problems, etc.

Soft constraints allow users to express their local preferences over constraints, variables and/or tuples. However, in more recent works (e.g. [1]) novel ways of eliciting preferences is considered whereby users express preferences over complete assignments. The challenge then becomes how to model such general preferences locally. Recall that the objective of solving problems with preferences is to find an optimal solution. For this reason, it is very important to acquire local preferences with the motivation to help find an optimal solution of the problem. In this context, our paper focuses in the local preferences acquisition task.

The Weighted Partial MaxSAT problem (WPMS) allows us to express local preferences since it incorporates weights associated with soft clauses. Then, by means of the weights associated to the soft constraints, the users can fix the preferences locally, over each soft constraint. Each weight is the penalty associated with a solution that does not satisfy the clause. The higher the weight of a clause, the higher is the preference of the satisfaction of that clause. The objective function is to find a solution to the hard clauses that minimizes the sum of the weights of the unsatisfied soft clauses.

We have approached this task by modeling the problem as a system of linear equations and subsequently using linear algebra techniques to solve the resultant system. Furthermore, we are interested in the characteristics and size of the complete assignments, and their total costs, required to acquire all local weights.

Preliminaries

The SAT and MaxSAT problems are well known to be NP-Hard [2, 3]. In SAT, for a given boolean formula, the objective is to find an assignment that satisfies all clauses in the formula. In MaxSAT the objective is to find an assignment that maximizes the number of satisfied clauses.

A CNF is a set of clauses $(C_1, C_2, ..., C_m)$ in conjunctive normal form where C_i is a disjunction of literals. A literal is a boolean variable x or its negation $\overline{x}$ [4]. The SAT problem is to find an assignment $var(f) \to \forall C_i \in f, C_i \to True$. Let ϕ be a function such that $\phi(C) \to 1$ iff $C \to True$, otherwise $\phi(C) \to 0$. The MaxSAT problem is to find an assignment that satisfies the maximum number of clauses: $var(f) \to max(\sum_1^m \phi(C_i))$.

The following notation was described in [5]. The weighted partial MaxSAT (WPMS) problem is composed of a set of hard clauses and a set of soft clauses, each with a weight. The weight associated with each soft clause C_i is denoted as w_i. A weighted clause is a pair (C_i, w_i), where w_i is natural number. The formula defining a WPMS problem is as follows:

$$\varphi = \{(C_1, w_1), .., (C_m, w_m), .., (C_{m+1}, \infty), .., (C_{m+m'}, \infty)\}$$

where the first m clauses are soft and the last m' are hard. The weight of hard clauses should be ∞ compared to soft ones. $var(\varphi), var(C)$ denotes the set of variables in φ and C respectively. I is a function that gives the truth assignment for literals, clauses, SAT formulates. Then, $I : var(\varphi) \to \{0, 1\}$ is noted as the assignment for φ. $cost(\varphi) = \sum_{i=1}^m w_i(1 - I(C_i))$. Each w_i is added to the total cost iff clause C_i is unsatisfied. The objective function of the WPMS is to minimize $cost(\varphi)$, i.e., to minimize the total cost associated with the set of unsatisfied clauses.

Formalization

We intend to acquire the unknown weights of the soft clauses of a WPMS problem. Without loss of generality and with the purpose of simplifying notation, we consider that all the weights of the m soft clauses of the WPMS are unknown. We use the fact that, by definition, the relation between w and $cost(\varphi)$ is linear.

For solving this system of linear equations we use Gaussian elimination [6, 7], which performs elementary row reductions by performing operations over the coefficient matrix. With the first part of the process of Gaussian elimination the original matrix of coefficients is converted into its row echelon form. The second part of the algorithm continues the row reduction until its convergence to a row reduced echelon form, in order to find the values of the undetermined variables, i.e. to find the solution of the system of linear equations.

Experiments

The objective of our experiments was to evaluate the effectiveness of our algorithm in finding the full set of m LI solutions and subsequently using them for acquiring the m unknown weights in a WPMS instance. In our experiments, we used SCIP as a SAT solver for finding the solutions that satisfy the hard clauses. As previously mentioned, after each iteration, we added or updated the clauses in the SAT instance in order to find a new LI solution. We set up a stop-condition to terminate the search: an iteration-limit $\#NoNewSolIters > \#SoftClaus - \#LISols$.

We performed our experiments on a set of benchmarks taken from the MaxSAT Evaluations competitions. These datasets are either from `random`, `crafted` or `industrial` subsets of the `Weighted Partial MaxSAT` catalog of MaxSAT Evaluations competitions 2006. Table 1 shows the benchmarks analyzed in ascending order by the number of total clauses in the dataset. Column `#Total Claus` represents the number of total clauses in the dataset. The hard clauses and soft clauses in the column `#Hard Clause` and `#Soft Clause` respectively. We assume that all the soft clauses have unknown weights. The number of variables is represented as `#Vars`. In the `Result` part of the Table 1, we list the number of LI solutions found, the number of acquired unknown weights, runtime and the number of iterations in experiment.

The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289

Table: Benchmarks and results obtained.

	Benchmark Details				Result			
Benchmark	#Total Claus	#Hard Claus	#Vars	#Soft Claus	#LI Sols	#ACQ Claus	Time(s)	#Iters
8.wcsp.log	25	17	12	8	7	**7**	<1	15
54.wcsp.log	479	412	96	67	**67**	**67**	171	**67**
cat_paths_60_80_0002.txt	553	472	81	81	**81**	**81**	390	**81**
cat_paths_60_70_0006.txt	566	495	71	71	**71**	**71**	142	**71**
29.wcsp.log	692	610	101	82	**82**	**82**	300	**82**
404.wcsp.log	1037	937	129	100	**100**	**100**	619	**100**
cat_paths_60_140_0002.txt	1833	1692	141	141	**141**	**141**	3436	143
cat_paths_60_140_0003.txt	1954	1810	144	144	**144**	**144**	5395	179
cat_paths_60_160_0001.txt	2377	2216	161	161	**161**	**161**	8247	242
cat_paths_60_150_0001.txt	2460	2310	150	150	**150**	**150**	5580	168
cat_paths_60_170_0000.txt	2803	2633	170	170	**170**	**170**	6328	177
cat_paths_60_170_0003.txt	3016	2845	171	171	**171**	**171**	7950	**171**
408.wcsp.log	3149	2949	264	200	**200**	**200**	18228	220
cap72.wcsp	3978	3164	1664	814	37	11	5370	853
1504.wcsp.log	6593	5988	929	605	232	2	176013	488

The results of the experiments performed show that for many benchmarks we were able to find the same number of LI solutions as the number of soft clauses. Therefore, in such cases, we were able to acquire all the unknown weights of all the soft clauses by using Gaussian elimination.

While the above experiments shows that we can acquire the m unknown associated weights for soft clauses if we have the full set of m LI solutions and the associated weights for solutions given by user, we are also interested in cases where a user can only roughly estimate the weights of solutions or give the order of solutions. We performed another set of experiments in which we applied some errors to the associated weights for solutions to see if the computed optimal solutions change drastically.

In the case that the user can only give the order of solutions, we could consider the given order as some rough weights, e.g. obtaining $S_1 > S_2 > S_3 > ...$ from $S_1 = 100, S_2 = 200, S_3 = 300, ...$.

In this set of experiments, we used the acquired weights for soft clauses to compute the optimal solution and its weight (denoted as O). After errors were applied into the weights of solutions, we acquired the new associated weights for soft clauses and computed the new optimal solution and its weight (denoted as O'). Then we computed the Δ between O' and O.

Table: Applied errors and the resulting delta of optimal solutions.

	not intentionally in maintaining the order of solutions						remain the order of solutions after applying errors as before					
	10%		20%		30%		10%		20%		30%	
Benchmark	ΔO'-O	Δ/O	ΔO'-O	Δ/O	ΔO'-O	Δ/O	ΔO'-O	Δ/O	ΔO'-O	Δ/O	ΔO'-O	Δ/O
29.wcsp.log	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**
404.wcsp.log	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**
54.wcsp.log	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**
8.wcsp.log	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**	**0**	**0%**
..60_140_0002..	139	0.150%	293	0.316%	472	0.509%	**0**	**0%**	**0**	**0%**	**0**	**0%**
..60_140_0003..	68	0.073%	192	0.207%	282	0.304%	**0**	**0%**	**0**	**0%**	**0**	**0%**
..60_150_0001..	73	0.079%	224	0.243%	400	0.433%	23	0.025%	21	0.023%	21	0.023%
..60_160_0001..	128	0.129%	86	0.087%	388	0.391%	5	0.005%	17	0.017%	17	0.017%
..60_170_0000..	58	0.047%	317	0.259%	434	0.355%	13	0.011%	13	0.011%	26	0.021%
..60_170_0003..	65	0.053%	183	0.149%	484	0.394%	**0**	**0%**	**0**	**0%**	8	0.007%
..60_70_0006..	63	0.143%	88	0.200%	175	0.397%	3	0.007%	4	0.009%	14	0.032%
..60_80_0002..	15	0.033%	23	0.051%	127	0.281%	**0**	**0%**	15	0.033%	16	0.035%

In this experiment, there are three error intervals of **10%**, **20%** and **30%** for each benchmark. Also, for each benchmark we considered two different options with respect to the order of solutions after applying errors. First relates to the order of solutions after applying error as before, the other does not intentionally maintain the order.

In Table 2 for each interval we present Δ O'-O and the percentage of Δ to the weight of optimal solution before applying errors (denoted as Δ/O). The results shown in Table 2 are the average result of 10 experiments.

The results show that if the user can only roughly estimate the weights of solutions, or just specify the order of solutions, which are similar to applying errors into the weights of solutions, we still can find a close solution to the optimal.

Future Work

- We will work on a more restrictive variant of the problem described in this paper in which we can only compute k solutions whose associated equations are linearly independent, where k < m. Therefore, we want to find the k solutions that maximize the number of weights acquired.

References

Rossi, F., Sperduti, A.: Acquiring both constraint and solution preferences in interactive constraint systems. Constraints 9(4), 311–332 (2004)

Bliznets, I., Golovnev, A.: A new algorithm for parameterized MAX-SAT. In: Parameterized and Exact Computation - 7th International Symposium, IPEC 2012, Ljubljana, Slovenia, September 12-14, 2012. Proceedings. pp. 37–48 (2012), `http://dx.doi.org/10.1007/978-3-642-33293-7_6`

Patrascu, M., Williams, R.: On the possibility of faster SAT algorithms. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010. pp. 1065–1075 (2010), `http://dx.doi.org/10.1137/1.9781611973075.86`

Eén, N., Biere, A.: Effective preprocessing in SAT through variable and clause elimination. In: Theory and Applications of Satisfiability Testing, 8th International Conference, SAT 2005, St. Andrews, UK, June 19-23, 2005, Proceedings. pp. 61–75 (2005), `http://dx.doi.org/10.1007/11499107_5`

Ansótegui, C., Gabàs, J.: Solving (weighted) partial maxsat with ILP. In: Proceedings of CPAIOR. pp. 403–409 (2013), `http://www.cis.cornell.edu/ics/cpaior2013/pdfs/ansotegui.pdf`

Bareiss, E.H.: Sylvester's identity and multistep integer-preserving Gaussian elimination. Mathematics of Computation 22(103), 565–578 (Jul 1968)

Gentle, J.E.: Gaussian elimination. In: Numerical Linear Algebra for Applications in Statistics, chap. 3.1, pp. 87–91. Springer-Verlag, pub-SV:adr (1998)

DCU NUI Galway OÉ Gaillimh UCC University College Cork, Ireland Coláiste na hOllscoile Corcaigh UCD DUBLIN sfi Science Foundation Ireland