MaxSAT-based Scheduling of B2B Meetings

Miquel Bofill*, Marc Garcia, Josep Suy*, and Mateu Villaret*

Departament d'Informàtica, Matemàtica Aplicada i Estadística
Universitat de Girona, Spain
{mbofill,mgarciao,suy,villaret}@imae.udg.edu

Abstract. In this work we propose a MaxSAT formulation for the problem of scheduling business-to-business meetings. We identify some implied constraints and provide distinct encodings of the used cardinality constraints. The experimental results show that the proposed technique outperforms previous existing approaches on this problem.

1 Introduction

In recent years, solving combinatorial problems by encoding them into propositional formulae and proving their satisfiability has been proved to be a robust approach. Although the field of automated timetabling is dominated by local search heuristic methods, SAT and MaxSAT-based methods have been successfully applied to curriculum-based timetabling problems recently [2,5].

In this work we propose some MaxSAT encodings for the problem of scheduling business-to-business (B2B) bilateral meetings. Those meeting sessions occur in events from several fields like sports, research, etc. and aim to facilitate participants with similar interests meeting each other. We consider the generation of timetables for B2B meetings in the particular setting of the *Forum of the UdG's Science and Technology Park*.[1] The goal of this forum is to be a technological marketplace in Girona, by bringing the opportunity to companies, research groups, investors, etc., to find future business partnerships.

In the scheduling of B2B meetings it is desirable to avoid unnecessary idle time periods between meetings, and to be fair in such minimization, i.e., to avoid big differences in the number of idle time periods among participants. Experience shows that idle time periods may led some participants to leave the event, dismissing later scheduled meetings. We propose to face this optimization problem by encoding it as a partial MaxSAT formula [11], where some clauses are marked as hard whilst others are marked as soft, and the goal is to find an assignment to the variables that satisfies all hard clauses and falsifies the minimum number of soft clauses. In our case, the falsification of a soft clause will represent the existence of an idle time period for some participant.

As far as we know, there are not many works dealing with this problem. In [7] we can find a system that is used by the company *piranha womex AG* for

* Supported by the Spanish Ministry of Science and Innovation (project TIN2012-33042).

[1] http://www.parcudg.com

computing matchmaking schedules in several fairs. That system differs from ours in some aspects: e.g., it considers neither forbidden slots nor fairness. In [3] the authors proposed a Constraint Programming (CP) model and a pseudo-Boolean (PB) model, and provided performance results for basic configurations of several solvers. Among the model and solver combinations considered, the CP model with a SMT solver, and the PseudoBoolean model with a SAT-based PB solver, were shown to be the most robust. This fact motivates us to go one step further with SAT-based technology by providing a direct MaxSAT encoding for this problem in this paper. We report on experiments showing that a basic MaxSAT encoding outperforms previous results. We show that even better results can be obtained with state-of-the-art encodings of cardinality constraints and some improvements and extensions of the initial encoding. As test suite we use (and provide) the industrial instances of previous editions of the Forum and some crafted modifications of those.

2 The B2B problem

Here we define the problem at hand. For more details on the nature of the problem and the instances from the Forum of the UdG's Science and Technology Park, see [3].

Definition 1 (B2BSOP-d). *Given a set of participants, a list of time slots, a set of available locations (tables) and a set of meetings between pairs of participants, where for each participant there can be forbidden time slots and meetings may be required to be held in morning or afternoon slots, the goal of the* "B2B Scheduling Optimization Problem with homogeneity d" *is to find a total mapping from the meetings to time slots and locations such that:*

- *At most one meeting involving the same participant is scheduled in each time slot.*
- *No meeting is scheduled in a forbidden time slot for any of its participants.*
- *Each meeting having a morning or afternoon slot requirement is scheduled in a time slot of the appropriate interval.*
- *The difference between the number of idle time periods of each pair of participants is at most d, where by an* idle time period *we refer to a group of idle time slots between two successive meetings involving the same participant.*
- *The total number of idle time periods is minimized.*

3 Encodings

3.1 MaxSAT Base Encoding for the B2BSOP-d

Parameters Each instance is defined by the following parameters.

nMeetings: number of meetings
nTimeSlots: number of available time slots

$nMorningSlots$: number of morning time slots
$nTables$: number of available locations
$nParticipants$: number of participants
$morningMeetings$: subset of $\{1, \ldots, nMeetings\}$ to be scheduled in morning slots
$afternoonMeetings$: subset of $\{1, \ldots, nMeetings\}$ to be scheduled in afternoon slots
$meetings$, function from $\{1, \ldots, nParticipants\}$ to $2^{\{1,\ldots,nMeetings\}}$: set of meetings involving each participant
$forbidden$, function from $\{1, \ldots, nParticipants\}$ to $2^{\{1,\ldots,nTimeSlots\}}$: set of forbidden time slots for each participant

Variables We define the following propositional variables.

$schedule_{i,j}$: meeting i is held in time slot j
$usedSlot_{p,j}$: participant p has a meeting scheduled in time slot j
$fromSlot_{p,j}$: participant p has a meeting scheduled at, or before, time slot j
$endHole_{p,j}$: participant p has an idle time period finishing at time slot j
$max_1, \ldots, max_{\lfloor (nTimeSlots-1)/2 \rfloor}$ and $min_1, \ldots, min_{\lfloor (nTimeSlots-1)/2 \rfloor}$: unary representation of an upper bound and a lower bound of the maximum and minimum number of idle time periods among all participants, respectively. Note that there can be at most $\lfloor (nTimeSlots - 1)/2 \rfloor$ idle time periods per participant. By restricting the difference between these variables to be less than a certain value, we will enforce homogeneity of solutions (Constraints (16) to (21)).

We also use some auxiliary variables that will be introduced when needed.

Constraints All constraints except (15) are hard. To help readability we define $M = \{1, \ldots, nMeetings\}$, $T = \{1, \ldots, nTimeSlots\}$, $P = \{1, \ldots, nParticipants\}$.

- *At most one meeting involving the same participant is scheduled in each time slot.*

$$atMost(1, \{schedule_{i,j} \mid i \in meetings(p)\}) \qquad \forall p \in P, j \in T \tag{1}$$

- *No meeting is scheduled in a forbidden time slot for any of its participants.*

$$\bigwedge_{i \in meetings(p),\, j \in forbidden(p)} \neg schedule_{i,j} \qquad \forall p \in P \tag{2}$$

- *Each meeting having a morning or afternoon slot requirement is scheduled in a time slot of the appropriate interval.*

$$exactly(1, \{schedule_{i,j} \mid j \in 1..nMorningSlots\}) \qquad \forall i \in morningMeetings \tag{3}$$

$$\neg schedule_{i,j} \qquad \begin{array}{l} \forall i \in morningMeetings \\ \forall j \in nMorningSlots + 1..nTimeSlots \end{array} \tag{4}$$

$$exactly(1, \{schedule_{i,j} \mid j \in nMorningSlots + 1..nTimeSlots\}) \quad \forall i \in afternoonMeetings \tag{5}$$

$$\neg schedule_{i,j} \qquad \begin{matrix} \forall i \in afternoonMeetings \\ \forall j \in 1..nMorningSlots \end{matrix} \tag{6}$$

$$exactly(1, \{schedule_{i,j} \mid j \in T\}) \qquad \forall i \in M \setminus (morningMeetings \cup afternoonMeetings) \tag{7}$$

– *At most one meeting is scheduled in a given time slot and location.*

$$atMost(nTables, \{schedule_{i,j} \mid i \in M\}) \qquad \forall j \in T \tag{8}$$

Note that with Constraints (3) to (8) we get a total mapping from the meetings to time slots and locations.

In order to be able to minimize the number of idle time periods we introduce channeling constraints between the variables *schedule*, *usedSlot* and *fromSlot*.

– *If a meeting is scheduled in a certain time slot, then that time slot is used by both participants of the meeting.*

$$schedule_{i,j} \rightarrow (usedSlot_{p_i^1,j} \wedge usedSlot_{p_i^2,j}) \qquad \forall i \in M, j \in T$$
$$\text{where } p_i^1 \text{ and } p_i^2 \text{ are the participants of meeting } i. \tag{9}$$

In the reverse direction, if a time slot is used by some participant, then one of the meetings of that participant is scheduled in that time slot.

$$usedSlot_{p,j} \rightarrow \bigvee_{i \in meetings(p)} schedule_{i,j} \qquad \forall p \in P, j \in T \tag{10}$$

– *For each participant p and time slot j, $fromSlot_{p,j}$ is true if and only if participant p has had a meeting at or before time slot j.*

$$\neg usedSlot_{p,1} \rightarrow \neg fromSlot_{p,1} \quad \forall p \in P \tag{11}$$

$$(\neg fromSlot_{p,j-1} \wedge \neg usedSlot_{p,j}) \rightarrow \neg fromSlot_{p,j} \quad \forall p \in P, j \in T \setminus \{1\} \tag{12}$$

$$usedSlot_{p,j} \rightarrow fromSlot_{p,j} \quad \forall p \in P, j \in T \tag{13}$$

$$fromSlot_{p,j-1} \rightarrow fromSlot_{p,j} \quad \forall p \in P, j \in T \setminus \{1\} \tag{14}$$

Optimization Minimization of the number of idle time periods is achieved by means of soft constraints (except for the case of a cardinality network based encoding that we describe in Subsection 3.3).

– [Soft constraints] *If some participant does not have any meeting in a certain time slot, but it has had some meeting before, then she does not have any meeting in the following time slot.*

$$(\neg usedSlot_{p,j} \wedge fromSlot_{p,j}) \rightarrow \neg usedSlot_{p,j+1} \quad \forall p \in P, j \in T \setminus \{nTimeSlots\} \tag{15}$$

We claim that, with these constraints, an optimal solution will be one having the least number of idle time periods. Note that, for each participant, each meeting following some idle time period increases the cost by 1.

Remark 1. If we were just considering optimization, Constraints (11) and (12) would not be necessary, since minimization of the number of idle time periods would force the value of $fromSlot_{p,j}$ to be false for every participant p and time slot j previous to the first meeting of p. However, since we are also seeking for homogeneity, these constraints are mandatory. Without them, the value of $fromSlot_{p,j}$ could be set to true for time slots j previous to the first meeting of p, inducing a fake idle time period in order to satisfy the (hard) homogeneity constraints defined below. Constraints (11) and (12) were missing by mistake in [3].

Homogeneity We reify the violation of soft constraints in order to count the number of idle time periods of each participant. This will allow us to find the maximum and minimum number of idle time periods among all participants, and to enforce homogeneity by bounding their difference.

- *$endHole_{p,j}$ is true if and only if participant p has an idle time period finishing at time slot j.*

$$endHole_{p,j} \leftrightarrow \neg\left((\neg usedSlot_{p,j} \wedge fromSlot_{p,j}) \rightarrow \neg usedSlot_{p,j+1}\right) \qquad \forall p \in P, j \in T \setminus \{nTimeSlots\} \quad (16)$$

- *$sortedHole_{p,1}, \ldots, sortedHole_{p,nTimeSlots}$ are the unary representation of the number of idle time periods of each participant p.*

$$sortingNetwork([endHole_{p,j} \mid j \in T], [sortedHole_{p,j} \mid j \in T]) \qquad \forall p \in P \quad (17)$$

- *$max_1, \ldots, max_{\lfloor (nTimeSlots-1)/2 \rfloor}$ and $min_1, \ldots, min_{\lfloor (nTimeSlots-1)/2 \rfloor}$ are (an approximation to) the unary representation of the maximum and minimum number of idle time periods among all participants, respectively.*

$$sortedHole_{p,j} \rightarrow max_j \qquad \forall p \in P, j \in 1..\lfloor (nTimeSlots-1)/2 \rfloor \quad (18)$$

$$\neg sortedHole_{p,j} \rightarrow \neg min_j \qquad \forall p \in P, j \in 1..\lfloor (nTimeSlots-1)/2 \rfloor \quad (19)$$

  Constraints (18) and (19) are not enough to ensure that the *max* and *min* variables exactly represent the maximum and minimum number of idle time periods among all participants. However, together with Constraints (20) and (21), they suffice to soundly enforce the required homogeneity degree.

- *The difference between the maximum and minimum number of idle time periods can be at most d (in our setting the chosen number was 2).*

$$dif_j \leftrightarrow min_j \; XOR \; max_j \qquad \forall j \in 1..\lfloor (nTimeSlots-1)/2 \rfloor \quad (20)$$

$$atMost(d, \{dif_j \mid j \in 1..\lfloor (nTimeSlots-1)/2 \rfloor\}) \quad (21)$$

3.2 Extended Encoding

Implied Constraints We have identified the following implied constraints.

- *The number of meetings of a participant p as derived from* $usedSlot_{p,j}$ *variables must match the total number of meetings of p.*

$$exactly(|meetings(p)|, \{usedSlot_{p,j} \mid j \in T\}) \qquad \forall p \in P \tag{22}$$

- *The number of participants having a meeting in a given time slot is bounded by twice the number of available locations.*

$$atMost(2 \times nTables, \{usedSlot_{p,j} \mid p \in P\}) \qquad \forall j \in T \tag{23}$$

Symmetry Breaking With respect to symmetry breaking, the model implicitly eliminates possible table symmetries, since only the number of tables occupied is considered. Unfortunately, removing time symmetries in the presence of participants' forbidden time slots seems not to be feasible. However, we can break some time symmetries when there are no forbidden time slots and the meetings have neither morning nor afternoon slot requirements (there are several instances with these characteristics). Note that since we are minimizing the number of idle time periods, we cannot soundly break time symmetries by simply fixing a priori an ordering of meetings. Instead, what we do is to force some ordering in the "matrix" of *usedSlot* variables as follows, assuming an even number of time-slots and the existence of a participant with an odd number of meetings.[2]

- *For some participant p with an odd number of meetings we force the number of meetings of p taking place in the first half of time slots to be odd.*

$$((\ldots(usedSlot_{p,1}\ XOR\ usedSlot_{p,2})\ \ldots)\ XOR\ usedSlot_{p,\lfloor nTimeSlots/2 \rfloor}) \tag{24}$$

3.3 Encoding of Global Constraints

The cardinality constraints stating that at most (*atMost*) or exactly (*exactly*) k of a given set of variables must be true have been encoded in several ways. Similarly for the *sortingNetwork* constraint, which corresponds to a sorting network on a set of Boolean variables.

Naïve Encoding

- $atMost(1, \_)$: quadratic number of pairwise mutex clauses.
- $exactly(1, \_)$: $atMost(1, \_)$ plus a clause (disjunction) with all the involved variables, for the "at least" part of the constraint.
- *sortingNetwork*: odd-even sorting network.
- $atMost(k, \_)$: naïve sequential unary counter [12].

[2] All instances considered are like this.

Cardinality Networks based Encoding

- $atMost(1, \_)$: quadratic number of pairwise mutex clauses.
- $exactly(1, \_)$: commander-variable encoding [9].
- $exactly(k, \_)$, *sortingNetwork* and $atMost(k, \_)$: cardinality networks [1].

By using cardinality networks we can deal with soft constraints in a more clever way: instead of soft constraints (15), we post as soft constraint the negation of each "output variable" $sortedHole_{p,j}$ of the *sortingNetwork* corresponding to constraint (17). This way, knowing that each participant will have at most $\lfloor (nTimeSlots - 1)/2 \rfloor$ idle time periods, we can reduce the number of soft constraints, as well as the number of $sortedHole_{p,j}$ variables of each participant, to a half.

4 Experiments, Conclusions and Future Work

In this section we compare the performance of the state-of-the-art MaxSAT solver *QMaxSat14.04auto-g3* [10] using the proposed base model and a naïve encoding of the global constraints, with the performance of the best known method and solver for each instance in [3]. We also show how using cardinality networks for the global constraints, and extending the model with implied constraints and symmetry breaking, we can significantly improve the solving time. We use the same nine instances that the authors used in [3], plus new eleven instances.[3] Among all there are five industrial instances (the ones without *craf* annotation); the rest have been crafted from those by increasing the number of meetings, reducing the number of locations and removing the forbidden time slots.

All experiments have been run using the default options of each solver, on Intel® Xeon™CPU@3.1GHz machines, under CentOS release 6.3, kernel 2.6.32. Table 1 summarizes the results obtained. Only instances named *tic* do not contain forbidden time slots nor morning and afternoon preferences, hence symmetry related experiments are only reported for those. Column named **best known** shows the best results obtained in [3], where *PB clasp* and *PB cplex* refer to a pseudo-Boolean model solved with clasp 3.1.0 [8] and IBM ILOG CPLEX 12.6, respectively, and *CP sbdd* refers to a CP like model solved with WSimply using shared BDD optimization [4] and Yices 1.0.33 [6] as SMT solver. Columns **naïve** and **cardinal** show the results using the naïve and cardinality network based encodings of our base model, respectively. Columns **imp1**, **imp2**, **imp1+2** and **imp1+2+sym** show the results for the cardinality networks based encoding using implied constraints 1, 2, both, and both with symmetry breaking, respectively. The three numbers below the names of each instance are: the ratio between the median of meetings per participant and *nTimeSlots*, the ratio between *nTables* and *nParticipants*, and the ratio between the number of meetings to schedule and the available slots ($nTables \times nTimeSlots$).

[3] All instances can be found in `http://imae.udg.edu/recerca/lap/simply/`. Results from [3] have been updated according to Remark 1.

Table 1. Solving time (in seconds) and optimum found (number of idle time periods) per instance and solver. TO stands for 2 hours timeout. For aborted executions we report the (sub)optimum found if the solver reported any.

instance	best known		naïve		cardinal		imp1		imp2		imp1+2		imp1+2+sym	
forum-13 (0.20, 0.40, 0.52)	153.1 (PB clasp)	0	24.8	0	20.5	0	13.4	0	25.2	0	18.3	0	-	-
forum-13crafb (0.24, 0.36, 0.66)	TO (CP sbdd)	12	1492.7	6	83.4	6	82.4	6	83.1	6	81.2	6	-	-
forum-13crafc (0.20, 0.34, 0.61)	TO (CP sbdd)	20	116.3	1	1872.4	1	1661.3	1	1800.5	1	1300.2	1	-	-
forum-14 (0.35, 0.56, 0.62)	TO (CP sbdd)	7	TO	-	431.2	2	349.1	2	409.2	2	240.2	2	-	-
forumt-14 (0.79, 0.90, 0.87)	-	-	21.1	5	8.0	5	8.5	5	11.9	5	10.2	5	-	-
forumt-14crafc (0.79, 0.83, 0.94)	-	-	148.9	5	32.7	5	28.8	5	33.1	5	31.5	5	-	-
forumt-14crafd (0.78, 0.83, 0.94)	-	-	84.9	4	32.4	4	26.6	4	37.1	4	35.5	4	-	-
forumt-14crafe (0.78, 0.80, 0.98)	-	-	TO	-	95.2	5	78.1	5	105.2	5	94.7	5	-	-
ticf-13crafa (0.21, 0.40, 0.52)	-	-	21.2	0	24.6	0	15.0	0	45.9	0	35.9	0	-	-
ticf-13crafb (0.51, 0.36, 0.66)	-	-	3866.1	3	118.3	3	117.3	3	111.3	3	114.2	3	-	-
ticf-13crafc (0.21, 0.34, 0.61)	-	-	309.4	1	574.2	1	562.3	1	416.9	1	432.3	1	-	-
ticf-14crafa (0.35, 0.56, 0.62)	-	-	TO	-	TO	-	1532.8	0	2044.1	0	1339.6	0	-	-
tic-12 (0.74, 1.00, 0.74)	0.2 (PB clasp)	0	0.2	0	0.2	0	0.3	0	0.2	0	0.2	0	0.4	0
tic-12crafc (0.74, 0.76, 0.97)	53.2 (PB cplex)	0	7.8	0	4.1	0	3.1	0	2.5	0	2.6	0	3.4	0
tic-13 (0.76, 0.89, 0.85)	3.0 (PB clasp)	0	18.4	0	5.9	0	4.1	0	4.6	0	4.2	0	5.7	0
tic-13crafb (0.80, 0.89, 0.87)	2.1 (PB clasp)	0	3.6	0	2.4	0	2.6	0	7.1	0	5.5	0	4.1	0
tic-13crafc (0.76, 0.80, 0.94)	TO (PB cplex)	4	TO	4	25.9	4	19.1	4	25.2	4	23.9	4	26.1	4
tic-14crafa (0.79, 0.90, 0.87)	-	-	30.0	0	16.3	0	10.2	0	24.4	0	16.4	0	14.2	0
tic-14crafc (0.79, 0.83, 0.94)	-	-	740.0	0	49.3	0	45.1	0	45.7	0	44.5	0	56.8	0
tic-14crafd (0.79, 0.83, 0.94)	-	-	190.7	0	35.2	0	47.9	0	32.5	0	34.9	0	53.2	0

From the results reported we can extract the following conclusions: a) our base MaxSAT model with a naïve encoding outperforms all approaches considered in [3]; b) the cardinality network based encoding of our base model outperforms the naïve encoding; c) using implied constraints is in general beneficial; d) when the amount of information provided by the implied constraints is elevated is when really pays off to use them: in particular, for implied constraint 1, this happens when the ratio between the median of meetings per participant and *nTimeSlots* is low; for implied constraint 2, this happens when the ratio between *nTables* and *nParticipants* is low ; e) the use of symmetry breaking seems not to really help (in fact we think that we need some more hard instances to appreciate its possible benefits).

As future work we plan to find some more implied constraints and to improve symmetry breaking. We also plan to develop a portfolio with all these encodings, and to deeply compare with other solving techniques.

References

1. I. Abío, R. Nieuwenhuis, A. Oliveras, and E. Rodríguez-Carbonell. A Parametric Approach for Smaller and Better Encodings of Cardinality Constraints. In *19th International Conference on Principles and Practice of Constraint Programming , CP 2013*, volume 8124 of *LNCS*, pages 80–96. Springer, 2013.
2. R. J. A. Achá and R. Nieuwenhuis. Curriculum-based course timetabling with SAT and MaxSAT. *Annals OR*, 218(1):71–91, 2014.
3. M. Bofill, J. Espasa, M. Garcia, M. Palahí, J. Suy, and M. Villaret. Scheduling B2B Meetings. In *20th International Conference on Principles and Practice of Constraint Programming, CP 2014*, volume 8656 of *LNCS*, pages 781–796. Springer, 2014.
4. M. Bofill, M. Palahí, J. Suy, and M. Villaret. Solving Intensional Weighted CSPs by Incremental Optimization with BDDs. In *20th International Conference on Principles and Practice of Constraint Programming, CP 2014*, volume 8656 of *LNCS*, pages 207–223. Springer, 2014.
5. E. Demirović and N. Musliu. Modeling High School Timetabling as Partial Weighted MaxSAT. In *LaSh 2014: The 4th Workshop on Logic and Search (a SAT / ICLP workshop at FLoC 2014), Vienna, Austria*, 2014.
6. B. Dutertre and L. de Moura. The Yices SMT solver. Tool paper available at `http://yices.csl.sri.com/tool-paper.pdf`, August 2006. Accessed 23 Nov 2014.
7. M. Gebser, T. Glase, O. Sabuncu, and T. Schaub. Matchmaking with Answer Set Programming. In *12th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR 2013*, volume 8148 of *LNCS*, pages 342–347. Springer, 2013.
8. M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. *clasp* : A Conflict-Driven Answer Set Solver. In *9th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR 2007*, volume 4483 of *LNCS*, pages 260–265. Springer, 2007.
9. W. Klieber and G. Kwon. Efficient CNF encoding for selecting 1 from N objects. In *Fourth Workshop on Constraints in Formal Verification, CFV*, 2007.
10. M. Koshimura, T. Zhang, H. Fujita, and R. Hasegawa. QMaxSAT: A Partial Max-SAT Solver. *Journal on Satisfiability, Boolean Modeling and Computation*, 8(1/2):95–100, 2012.
11. C. M. Li and F. Manyà. *Handbook of Satisfiability*, chapter MaxSAT, Hard and Soft Constraints, pages 613–631. IOS Press, 2009.
12. C. Sinz. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. In *11th International Conference on Principles and Practice of Constraint Programming, CP 2005*, volume 3709 of *LNCS*, pages 827–831. Springer, 2005.