A Study on Implied Constraints in a MaxSAT Approach to B2B Problems

Miquel BOFILL [a], Jesús GIRÁLDEZ-CRU [b], Josep SUY [a] and Mateu VILLARET [a]
[a] *IMAE, Universitat de Girona*
e-mail: {mbofill,suy,villaret}@imae.udg.edu
[b] *DaSCI, DECSAI, Universidad de Granada*
e-mail: jgiraldez@ugr.es

Abstract. The B2B Scheduling Optimization Problem (B2BSOP) consists in finding an schedule of a set of meetings between pairs of participants, minimizing the number of idle time periods between meetings. Recent works have shown that SAT-based approaches are state-of-the-art on this problem. One interesting feature of such approaches is the use of implied constraints. In this work we provide an experimental setting to study the effectiveness of using these implied constraints. Since there is a reduced number of real-world B2B instances, we propose a random B2B instances generator, which reproduces certain features of the known real-world B2B instances. We show the usefulness of some implied constraints depending on the characteristics of the problem, and the benefits of combining them.

Keywords. timetabling, MaxSAT, implied constraints

1. Introduction

Business-to-business (B2B) events consist in meetings between people with similar interests. They are typically held in business networking events. The B2B scheduling problem is the problem of finding a feasible schedule for a set of requested meetings between pairs of participants, subject to participants' availability and accommodation capacity. To the best of our knowledge, there are a few works dealing with this problem [6,4,10,5].

The B2B scheduling optimization problem (B2BSOP) considered in this work was introduced in [4]. It consists in finding a feasible B2B schedule that minimizes the number of breaks (or idle time periods) in the participants' schedule. Side constraints can be added to ensure fairness among participants, e.g., restricting the maximum difference in the number of breaks between participants' schedules. A further refinement could consist in (additionally or alternatively) minimizing the duration of the breaks.

The B2BSOP has been successfully addressed by means of relatively straightforward CP and Pseudo-Boolean formulations [4], and further investigated in [10] with specialized CP and MIP formulations, and in [5] with partial MaxSAT specialized encodings. In a partial MaxSAT formula [8], some clauses are marked as hard whereas 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 the encoding of [5], which we will use throughout this paper, the falsification of a soft clause represents the existence of an idle time period for some participant. This MaxSAT encoding clearly dominates the other existing formulations on real-world instances, in which, moreover, the addition of implied constraints appears to be beneficial.

There exist some works in the literature showing the benefits of using implied constraints or redundant encodings in SAT [7,2,3]. However, the lack of varied real-world B2B instances makes difficult to hold this claim for the problem at hand. In this paper, we try to overcome this limitation by providing a parameterized random generator of B2B problem instances. The development of random problem generators sharing the majority of real-world instance features was already stated in [11] as one of the most important challenges in propositional search. The main goal of these generators is two-sided. On the one hand, generators allow to increment the set of instances since the generated instances share the main characteristics of the known instances. On the other hand, if they are correctly parameterized, they allow us to create instances with any desired property.

In this work we present a generator of random B2B instances, which is parameterized in several ways in order to reproduce certain characteristics of real-world B2B problems. This allows us to get some insights on the strengths and weaknesses of implied constraints, depending on the characteristics of the problem. In particular, we show the benefits of using certain implied constraints depending on the *density* and the *shape* of the problem. By *density* we refer to the ratio between the number of meetings and the accommodation capacity. By *shape* we refer to the configuration of the accommodation capacity, i.e., the ratio between timeslots and locations.

2. Definitions

In this section we provide a brief description of the B2B problem (defined in [4,5]). Due to space constraints, we do not provide a complete description of the MaxSAT encodings considered in the experiments. For a full description, we address the reader to [5].

Definition 1 (B2BSOP-*h*) *Given a set of participants $\mathcal{P}$, a list of timeslots $\mathcal{T}$, a set of available locations $\mathcal{L}$ and a set of meetings $\mathcal{M} \subset \mathcal{P} \times \mathcal{P}$ between pairs of participants to be scheduled, where participants may have forbidden meeting timeslots and celebrations of some meetings may be restricted to subsets of timeslots, the* B2B Scheduling Optimization Problem with homogeneity *h consists in finding a total mapping from $\mathcal{M}$ to $\mathcal{T} \times \mathcal{L}$, minimizing the total number of idle time periods and such that:*

(i) *Each participant has at most one meeting scheduled in each time slot.*
(ii) *No meeting is scheduled in a forbidden timeslot for any of its participants.*
(iii) *At most one meeting is scheduled in a given timeslot and location.*
(iv) *Each meeting restricted to a subset of timeslots (e.g., morning or afternoon meetings) is scheduled in one of those timeslots.*
(v) *The difference between the number of idle time periods among all participants is at most h, where by an* idle time period *we refer to a group of consecutive idle timeslots between two consecutive meetings involving the same participant.*

We use the MaxSAT encoding based on cardinality networks presented in [5]. In this encoding, among others, there are $|\mathcal{M}| \cdot |\mathcal{T}|$ Boolean variables $schedule_{i,j}$ representing if meeting i is held in timeslot j, and $|\mathcal{P}| \cdot |\mathcal{T}|$ Boolean auxiliary variables $usedSlot_{p,j}$ representing if participant p has a meeting scheduled in timeslot j. Using these variables, as well as other auxiliary variables and global constraints like $exactly(k,S)$ or $atMost(k,S)$ (stating that exactly k or at most k variables in a set S are set to true, respectively), it is possible to represent the constraints aforementioned. We remark that in this MaxSAT model the precise meeting celebration location is not considered. Instead, the amount of meetings celebrated per timeslot is bounded by the amount of available locations.

The minimization of the number of idle time periods is achieved by means of soft constraints. An idle time period can be identified when a participant has no meeting in a certain timeslot, has a meeting in the next timeslot, and has had a meeting before.

Additionally, the encoding incorporates some constraints in order to get a fair schedule. More precisely, there are homogeneity constraints enforcing that the difference between the participants' maximum and minimum number of idle time periods is bounded by the given parameter h.

Implied Constraints. The following two sets of implied constraints on *usedSlot* variables are identified:

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

$$exactly(|meetings(p)|, \{usedSlot_{p,j} \mid j \in \mathcal{T}\}) \;\; \forall p \in \mathcal{P} \tag{1}$$

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

$$atMost(2 \times nLocations, \{usedSlot_{p,j} \mid p \in \mathcal{P}\}) \;\; \forall j \in \mathcal{T} \tag{2}$$

Notice that these two implied constraints are not strictly necessary, hence they are redundant. Since every meeting has to be scheduled, these two constraints can be satisfied just using the variables *schedule*. In fact, in the original encoding [5], there is for instance a constraint limiting the number of scheduled meetings in the same timeslot to be less or equal than the number of available locations. Additionally, we derive these implied constraints with the variables *usedSlot*. The first implied constraint can be derived since no participant has more than one meeting in each timeslot, and the second implied constraint can be derived since at most one meeting is scheduled in every timeslot and location.

These constraints can be encoded in several ways into SAT, being *cardinality networks* [1] the most promising approach for this problem, according to the experiments in [5]. Moreover, as shown by the experiments, using these implied constraints is in general beneficial. However, it is difficult to extract conclusions from the experiments due to the limited number and variety of the available B2B instances.

3. A Random B2B Instances Generator

In this section we present a model for the generation of random B2B problems. To model these problems, we consider a number of participants $P = |\mathcal{P}|$ and a number of meetings $M = |\mathcal{M}|$. A key question is how these M meetings are distributed among these P participants.

The set of B2B instances provided in [5] consists of 20 instances: 5 instances obtained from real-world data, plus 15 crafted instances derived from the real ones. As we cannot draw general conclusions about probability distributions from this reduced set of instances, we present a simple model to generate random B2B instances. This model, called *B2Brand*, is based on an uniform probability distribution of meetings between pairs of participants in $\mathcal{P}$.

Definition 2 (B2Brand) *Let P be a positive number of participants with $P > 1$, and M, T and L positive numbers of meetings, timeslots, and locations, respectively. A random B2B instance from the B2Brand model is a set of M distinct meetings, independently selected by randomly choosing with uniform probability two distinct participants, whose identifiers range from 1 to P. Additionally, the instance must satisfy the following necessary feasibility conditions:*

- *The number of meetings M is bounded by the maximum combinations of two participants:*

$$M \leq \binom{P}{2} \tag{3}$$

- *The number of meetings M is bounded by the event capacity:*

$$M \leq T \cdot L \tag{4}$$

- *Each participant can have at most T meetings:*

$$|meetings(p)| \leq T \quad \forall p \,.\, 1 \leq p \leq P \tag{5}$$

where $|meetings(p)|$ is the number of meetings of participant p.

- *The number of meetings M is bounded by the combinations of participants and timeslots:*

$$M \leq \frac{P \cdot T}{2} \tag{6}$$

Notice that the last condition can be derived from Eq. 5 by summing up the number of meetings of every participant.

The model is parametric in the number of participants P and in the number of meetings M. However, it also needs the number of timeslots T and locations L in order to check the feasibility conditions defined. Notice that if any of those conditions is not satisfied, the B2B instance is trivially unsatisfiable. In particular, we define these feasibility conditions within the model because if the number of meetings is close to $\binom{P}{2}$ but $T \ll P$, then there would exist (with high probability) a number of participants requesting more meetings than available timeslots, and hence it would not be possible to schedule all of them. Forbidding this case, the extreme instance with $M \approx P \cdot T/2$ and $P \gg T$ will contain many participants requesting T meetings, i.e., many observations close to the maximum. In our experience, we have found this kind of participant to be very frequent in the real instances from [5].

Note also that the previous feasibility conditions do not guarantee the instance to be feasible. Let us consider for example a B2B instance with 3 participants, the 3 possible meetings between them, 2 timeslots, and 2 locations. In this problem, the four feasibility conditions are satisfied. However, it is easy to see that the instance is unfeasible.

In what follows, we use this model to generate random families of B2B instances, and we study the performance of the solving process on the use of the implied constraints. This study aims to check whether it is beneficial to use them or not, and in which situations it is more useful to use one or another. In particular, we focus our study on two features of the problem: the *density* and the *shape*.

Definition 3 (Density of a B2B instance) *Given a positive number of meetings M, timeslots T and locations L, the density d of a B2B instance is the relation between the number of meetings M and the accommodation capacity $T \cdot L$,*

$$d = \frac{M}{T \cdot L} \tag{7}$$

Notice that if the density of a B2B instance is greater than one, the instance is necessarily unfeasible. The opposite is not true. For example, the density of the instance mentioned before with 3 participants, the 3 possible meetings between them, 2 timeslots, and 2 locations is $d = 3/4$, but this instance is unfeasible. In fact, increasing the number of locations, i.e., decreasing the density, does not modify its feasibility. Therefore, unfeasible instances with density smaller than one do exist.

Definition 4 (Shape of a B2B instance) *Given the accommodation capacity $T \cdot L$, the shape s of a B2B instance is defined as the relation between the number of timeslots T and the number of locations L,*

$$s = \frac{T}{L} \tag{8}$$

Finally, we remark that this model does not represent all features of real-world instances. For instance, using this model we cannot generate a large number of participants requesting a number of meetings far from the mean. In particular, there could exist *passive* participants in some B2B events. These participants are characterized by requesting no meetings (i.e, they attend the event because other participants request meetings with them). In this case, using a different probability distribution to randomly select the two participants of each meeting, rather than uniform, would probably be a more suitable choice. However, our model seems adequate to model B2B instances similar to the known real-world ones. In fact, preliminary results on these existing B2B real-world instances (not presented in this paper for space constraints) hold with the conclusions drawn from our experimental evaluation, using the B2Brand model with this uniform probability distribution. In future works, we plan to extend our analysis to other probability distributions to model the number of meetings of every participant.

Another simplification of the B2Brand model is related with secondary requirements of the problem, namely, forbidden timeslots and fixed sessions for each meeting (by fixed sessions, we mean subsets of timeslots, e.g. morning or afternoon meetings). In random B2B instances generated with the B2Brand model, no meeting has any of these two constraints. By using this simplified model, we pursue to avoid any possible bias in our analysis possibly introduced by these secondary constraints.

4. Experimental Evaluation

In this section we evaluate the effect of using implied constraints in the solver performance. To this purpose, some families of random B2B instances have been generated[1], and solved with and without implied constraints. The goal of these experiments is to identify those cases for which the use of some implied constraint is beneficial.

Each of the generated families of random B2B benchmarks contains 10 instances per configuration. 36 different configurations have been used, resulting in a total of 1440 different random B2B instances. All experiments have been run with a timeout of 2 hours (7200 seconds). As in [5], a value of homogeneity $h = 2$ has been used.[2] Each instance has been solved without using implied constraints, as well as with implied constraint 1 (Eq. 1), implied constraint 2 (Eq. 2) and both. In the plots, these methods are named as `noimp`, `imp1`, `imp2` and `imp12`, respectively. Open-WBO [9] has been used as MaxSAT solver. This solver has been ranked as one of the best non-portfolio solvers in the industrial partial MaxSAT tracks of the last MaxSAT Evaluations.[3] The experiments have been run in a cluster of nodes Intel Xeon E3-1220v2 at 3.10 GHz with 8GB of RAM.

In our analysis, we represent the Penalized Average Runtime 1 (PAR1), which is the average of the runtime used to solve the set of instances in a family, assigning to unsolved instances the used timeout (i.e., 7200 seconds). For each family, we depict a box-and-whisker plot, which represents the maximum, minimum, median, and quartiles 1 and 3 of the runtimes of the family.

Based on the definitions of density and shape, we can easily create families of random B2B problems by setting different values to the parameters of the B2Brand model. In all the experiments, the random problems have $P = 40$ participants, which is a realistic number of participants according to existing real-world B2B instances. In a first batch of experiments, we analyze the hardness of random B2B problems with respect to their density. To this purpose, we fix the number of timeslots T and the number of locations L, and generate several families of B2B random instances varying the number of meetings M. We do the same experiment for four different combinations of T and L (i.e., four different shapes), where the product $T \cdot L$ is the same. Therefore, we represent the same densities in the four plots (since we evaluate the same values of M and the product $T \cdot L$ is the same in the four plots). In particular, we generate the following B2B random instances: $P = 40$, $M = \{70, 90, 110, 130, 150\}$, and (i) $T = 20$ and $L = 8$, (ii) $T = 16$ and $L = 10$, (iii) $T = 10$ and $L = 16$, and (iv) $T = 8$ and $L = 20$. Figure 1 shows the results of this experiment.

We first observe that the lower the density is (i.e., the lower number of meetings M), the easier the instance becomes for the four encodings. This phenomenon is more clear as the shape gets higher (i.e., with a high number of timeslots T w.r.t. the number of locations L). This happens in the four shapes analyzed. This is expected since finding the optimum schedule of a lower number of meetings seems to be an easier task.

A second observation, related with the encodings, is that the encoding without implied constraints `noimp` always performs worse than the other three, whereas the encoding with both implied constraints `imp12` always shows the best performance.

[1] The generator can be found at `https://www.ugr.es/~jgiraldez/`

[2] We have also tested, instead of using the homogeneity (i.e., the maximum difference between the number of idle periods of every two partipants), to use an upper-bound on that number. We did not observe any difference in the optimums or in the solver performance, so we skip those experiments in our presentation.

[3] `http://www.maxsat.udl.cat/`.

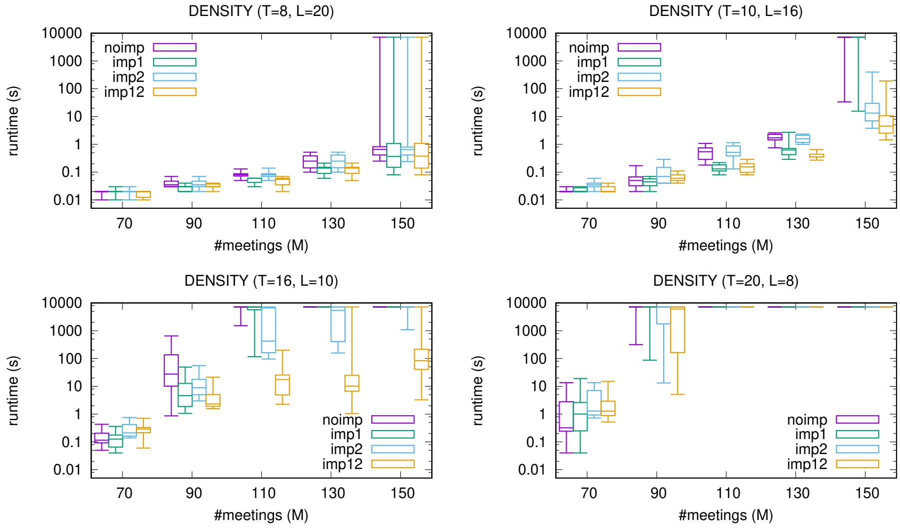

Figure 1. PAR1 (in seconds) of solving some random B2B families of instances, with and without using implied constraints, varying their density d. Instances are generated using the B2Brand model with $P = 40$, $M = \{70, 90, 110, 130, 150\}$, and (i) $T = 8$ and $L = 20$ (top left), (ii) $T = 10$ and $L = 16$ (top right), (iii) $T = 16$ and $L = 10$ (bottom left), and (iv) $T = 20$ and $L = 8$ (bottom right).

Interestingly, for the encodings with only one implied constraint (`imp1` and `imp2`), we can distinguish some cases where one performs better than the other. In particular, when the density is low, the encoding with the first implied constraint `imp1` seems to perform faster. See, for instance, the case with $M = 70$, $T = 16$ and $L = 10$ (bottom left plot in Fig. 1), for which `imp1` is clearly faster than `imp2`. On the contrary, when the density is high, `imp2` seems to outperform `imp1`. See, for instance, the case with $M = 150$, $T = 10$ and $L = 16$ (top right plot in Fig. 1), where `imp2` solves the 10 instances of the family whilst `imp1` only solves one. We summarize these results in Observation 1.

Observation 1 *Using the implied constraint 1 (`imp1`) seems to be more interesting when the density is small. On the contrary, the implied constraint 2 (`imp2`) seems to be more relevant when the density is high. Overall, using both implied constraints (`imp12`) is always beneficial.*

In a second batch of experiments, we analyze the hardness of random B2B problems with respect to their shape. To this purpose, we fix the number of meetings M and the number of timeslots T, and generate several families of B2B random instances varying the number of locations L. We do the same experiment for four different values of M (i.e., four different densities). Notice that we represent the same shapes in the four plots. In particular, we generate the following formulas: $P = 40$, $T = 20$, $L = \{8, 12, 16, 20\}$, and (i) $M = 80$, (ii) $M = 100$, (iii) $M = 130$, and (iv) $M = 150$. Figure 2 shows the results of this experiment.

We observe that the higher the density is (i.e., the lower number of locations L), the harder the B2B instance becomes. This happens in the four cases analyzed, although it occurs more clearly as the density gets higher (i.e., with a high number of meetings M). This is expected since reducing the number of available locations reduces the possible

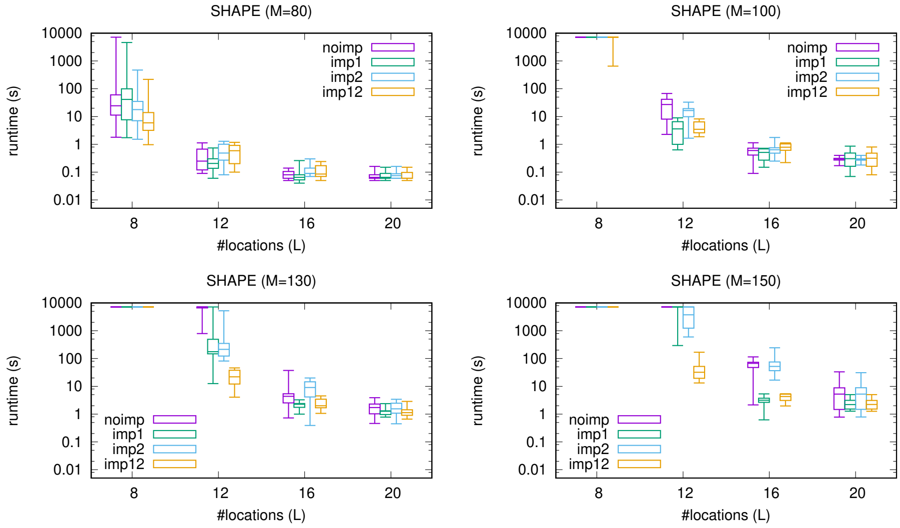

Figure 2. PAR1 (in seconds) of solving some random B2B families of instances, with and without using implied constraints, varying their shape s. Instances are generated using the B2Brand model with $P = 40$, $T = 20$, $L = \{8, 12, 16, 20\}$, and (i) $M = 80$ (top left), (ii) $M = 100$ (top right), (iii) $M = 130$ (bottom left), and (iv) $M = 150$ (bottom right).

parallelism among meetings, possibly creating some idle periods for some participants. Thus, finding the optimum schedule may be a harder task.

Again, using no implied constraints `noimp` is always worse than using any, whereas using both `imp12` is always better than the other three encodings.

As in the previous analysis, in this second analysis of B2B random instances varying their shape it is interesting to observe that some configurations, where using only one implied constraint performs better than using the other, can also be distinguished. In particular, when the shape is high, the encoding with the first implied constraint `imp1` seems to perform faster. See, for instance, the case with $L = 16$ and $M = 130$ (bottom left plot in Fig. 2), for which `imp1` is clearly faster than `imp2`. On the contrary, when the shape is low, `imp2` seems to outperform `imp1`. See, for instance, the case with $L = 12$ and $M = 150$ (bottom right plot in Fig. 2), where `imp2` solves most of the B2B instances of the family, but `imp1` only solves one. We summarize these results in Observation 2.

Observation 2 *Using the implied constraint 1 (`imp1`) seems to be more interesting when the shape is high. On the contrary, the implied constraint 2 (`imp2`) seems to be more relevant when the shape is low. Overall, using both implied constraints (`imp12`) is always beneficial.*

The intuition behind these observations may be connected to the relation between the implied constraints and the number of locations and timeslots. In particular, the first implied constraint depends on the number of meetings of each participant, and this number is bounded by the number of timeslots. The second implied constraint depends on the number of locations. Notice that since we are using cardinality networks to encode these constraints, the smaller the number of timeslots is, the better for the encoding of the first implied constraint, and similarly with the number of locations and the second implied constraint. Overall, using both implied constraints seems to find and propagate contradictions faster either when a participant has been scheduled more meetings than his/her meetings (i.e., implied constraint 1), or when a timeslot has been scheduled more meetings than the number of available locations (i.e., implied constraint 2).

5. Conclusions and Future Work

We have provided an experimental study of the effectiveness of using implied constraints in B2B scheduling problems using MaxSAT-based encodings.

Due to the limited number of real-world instances, we have proposed a random B2B instances generator, using a model based on an uniform probability of a participant requesting a meeting with another. Using the generator, we have generated families of random B2B instances, and studied the strengths and weaknesses of using implied constraints depending on the characteristics of the instance. We have focused the analysis on the density (i.e., the ratio between the number of meetings and the accommodation capacity) and the shape (i.e., the configuration of the accommodation capacity).

We have observed that there is a duality in the benefits of using the two implied constraints studied in this work. For small densities or high shapes, we have seen that it is more useful to use the implied constraint `imp1`. On the contrary, for high densities or low shapes, the second implied constraint `imp2` is more beneficial. Overall, the use of both implied constraints results into a very good performance in all cases.

As future work, we propose to extend this analysis in two different directions. On the one hand, we plan to extend our random B2B model in order to incorporate *passive* participants. As stated before, this can be achieved by using a different probability distributions when choosing the two participants of each meetings, rather the uniform distribution used in the current model. Additionally, the random model can be extended introducing secondary constraints, like forbidden timeslots and fixed sessions (e.g., morning and afternoon meetings). On the other hand, we propose to study solver heuristics that, for instance, prioritize taking decisions on the most relevant variables of the model (e.g., the variables that encode the implied constraints `imp1` or `imp2`), depending on the characteristics of the problem, as the density or the shape.

Acknowledgements

This work is partially supported by grants TIN2015-66293-R (MINECO/FEDER, UE), PGC2018-101216-B-I00 and RTI2018-095609-B-I00 (MICINN/FEDER, UE) and fellowship FJCI-2017-32420 (MICINN/Juan de la Cierva).

References

[1] Ignasi Abío, Robert Nieuwenhuis, Albert Oliveras, and Enric 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] Teresa Alsinet, Ramón Béjar, Alba Cabiscol, Cèsar Fernández, and Felip Manyà. Minimal and redundant SAT encodings for the all-interval-series problem. In *5th Catalonian Conference on Topics in Artificial Intelligence CCIA 2002*, pages 139–144, 2002.

[3] Carlos Ansótegui, Alvaro del Val, Iván Dotú, Cèsar Fernández, and Felip Manyà. Modeling choices in quasigroup completion: SAT vs. CSP. In *19th National Conference on Artificial Intelligence, AAAI 2004*, pages 137–142, 2004.

[4] Miquel Bofill, Joan Espasa, Marc Garcia, Miquel Palahí, Josep Suy, and Mateu 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.

[5] Miquel Bofill, Marc Garcia, Josep Suy, and Mateu Villaret. MaxSAT-Based Scheduling of B2B Meetings. In *12th International Conference on Integration of AI and OR Techniques in Constraint Programming, CPAIOR 2015*, volume 9075 of *LNCS*, pages 65–73. Springer, 2015.

[6] Martin Gebser, Thomas Glase, Orkunt Sabuncu, and Torsten Schaub. Matchmaking with answer set programming. In *12th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR 2013*, pages 342–347, 2013.

[7] Henry A. Kautz, Yongshao Ruan, Dimitris Achlioptas, Carla P. Gomes, Bart Selman, and Mark E. Stickel. Balance and filtering in structured satisfiable problems. In *17th International Joint Conference on Artificial Intelligence, IJCAI 2001*, pages 351–358, 2001.

[8] Chu Min Li and Felip Manyà. *Handbook of Satisfiability*, chapter MaxSAT, Hard and Soft Constraints, pages 613–631. IOS Press, 2009.

[9] Ruben Martins, Vasco M. Manquinho, and Inês Lynce. Open-WBO: A modular MaxSAT solver,. In *17th International Conference on Theory and Applications of Satisfiability Testing, SAT 2014*, pages 438–445, 2014.

[10] Gilles Pesant, Gregory Rix, and Louis-Martin Rousseau. A Comparative Study of MIP and CP Formulations for the B2B Scheduling Optimization Problem. In *12th International Conference on Integration of AI and OR Techniques in Constraint Programming, CPAIOR 2015*, volume 9075 of *LNCS*, pages 306–321. Springer, 2015.

[11] Bart Selman, Henry A. Kautz, and David A. McAllester. Ten challenges in propositional reasoning and search. In *15th International Joint Conference on Artificial Intelligence, IJCAI 97*, pages 50–54. Morgan Kaufmann, 1997.