

Planification des sessions parallèles de la conférence ROADEF via la satisfiabilité maximum

Sami Cherif¹

Heythem Sattoutah^{1,2}

Chu-Min Li¹

Corinne Lucet¹

Laure Brisoux-Devendeville¹

¹ Laboratoire MIS UR 4290, Université de Picardie Jules Verne, Amiens, France

² École supérieure en informatique de Sidi Bel Abbès, Algérie

{prenom.nom}@u-picardie.fr

Résumé

Cet article explore l'application de la Satisfiabilité Maximum comme paradigme d'optimisation pour résoudre le problème de planification de sessions de conférences de manière efficace. Une attention particulière est portée sur la minimisation des conflits entre les groupes de travail dans le contexte de la conférence ROADEF. Les résultats des solveurs de la littérature sur les instances de ROADEF 2024 démontrent la capacité de notre modèle à planifier efficacement les sessions, tout en permettant de réduire le nombre de conflits ainsi que le nombre maximum de sessions parallèles par rapport à la solution manuelle proposée par le comité d'organisation.

Abstract

This paper explores the application of Maximum Satisfiability to the complex problem of conference session scheduling, with a particular focus on minimizing working-group conflicts within the context of the ROADEF conference. The results of applying our formulation to data from ROADEF 2024 demonstrate the ability of our model to effectively compute session schedules in a short time frame, while enabling to reduce the number of conflicts as well as the maximum number of parallel sessions compared to the handmade solution proposed by the organizing committee.

1 Introduction

Le Problème de Satisfiabilité Maximum (Max-SAT) est une extension naturelle en problème d'optimisation de la Satisfiabilité propositionnelle (SAT) [4]. Le problème SAT consiste simplement à vérifier si toutes les contraintes clausales dans une formule donnée en entrée peuvent être satisfaites par une interprétation des variables, tandis que Max-SAT détermine le nombre maximal de contraintes pouvant être satisfaites par assignation des variables dans la formule.

La dernière décennie a connu des avancées majeures dans la résolution de Max-SAT. En effet, les algorithmes basés sur SAT [1, 10, 13] sont capables d'exploiter la puissance des solveurs SAT modernes et, en particulier, leur capacité à calculer efficacement des *cores*, c'est-à-dire des sous-ensembles insatisfiables de la formule. Plus récemment, les algorithmes de type séparation et évaluation pour Max-SAT ont également connu des progrès considérables grâce à l'introduction d'un processus d'apprentissage dédié [12]. Ces avancées ont fait de Max-Sat un formalisme puissant pouvant être utilisé pour modéliser et résoudre efficacement de nombreux problèmes académiques et industriels. En particulier, Max-SAT a été utilisé pour aborder des applications dans divers domaines tels que la planification [2, 6, 7, 8], la sécurité [17], le routage [14], l'analyse de données [3] ou encore la bio-informatique [11], parmi tant d'autres.

Dans cet article, nous abordons un problème de planification de conférence à travers le prisme de Max-SAT. Notre objectif est de résoudre un problème survenu lors de l'organisation de la conférence nationale ROADEF, le plus grand événement francophone visant à réunir des chercheurs des communautés de la recherche opérationnelle et de l'optimisation combinatoire. Pour cela, nous introduisons un modèle Max-SAT permettant de minimiser les conflits entre groupes de travail dans la planification des sessions parallèles pendant la conférence. Nous montrons que notre méthode offre une solution systématique et efficace aux défis complexes associés à notre problème, garantissant une expérience plus fluide et plus productive pour les participants et les organisateurs, et permettant notamment de réduire le nombre de conflits ainsi que le nombre maximal de sessions parallèles par rapport à la solution manuelle proposée par le comité d'organisation.

2 Préliminaires

Soit V l'ensemble des variables propositionnelles. Un littéral l est une variable $x \in V$ ou sa négation $\bar{x}$. Une clause est une disjonction de littéraux. Une formule en forme normale conjonctive (CNF) est une conjonction de clauses. Une interprétation $\alpha : V \rightarrow \{true, false\}$ associe à chaque variable une valeur booléenne et peut être représentée comme un ensemble de littéraux. Un littéral l est satisfait par une assignation α si $l \in \alpha$, sinon il est falsifié par α . Une clause est satisfaite par une interprétation α si au moins l'un de ses littéraux est satisfait par α , sinon elle est falsifiée par α . Le coût d'une interprétation α , noté $cost_\alpha(\phi)$, est le nombre de clauses dans la formule ϕ falsifiées par α . Résoudre le problème de Max-SAT consiste à déterminer le nombre maximal de clauses pouvant être satisfaites par une assignation de variables d'une formule CNF ϕ donnée en entrée ou, de manière équivalente, le nombre minimal de clauses que chaque interprétation doit falsifier, c'est-à-dire $opt(\phi) = \min_\alpha cost_\alpha(\phi)$. Le problème Max-SAT partiel est une variante bien connue prenant en entrée une formule CNF partielle, c'est-à-dire un ensemble biparti de clauses $\phi = H \cup S$ où H est l'ensemble des clauses dures qui doivent être satisfaites et S est l'ensemble des clauses souples à optimiser comme dans Max-SAT classique. L'objectif devient alors de calculer $opt(S)$ de manière à ce que toutes les clauses de H soient satisfaites. Dans nos modèles Max-SAT, nous utilisons également des contraintes pseudo-boléennes (PB) qui imposent une borne sur une somme pondérée de littéraux comme suit $(\sum_{i=1}^h a_i \times l_i) \circ k$ où $a_i, k \in \mathbb{N}$ et $\circ \in \{\leq, =, \geq\}$. Dans le cas où tous les littéraux de la somme sont des variables de V et que tous les poids sont fixés à 1 (et peuvent donc être omis), de telles contraintes sont plus couramment appelées contraintes de cardinalité. Les contraintes PB et de cardinalité peuvent être efficacement encodées en forme CNF [16] et sont typiquement utilisées dans les solveurs Max-SAT pour imposer des bornes pertinentes lors de la recherche.

3 Planification de la ROADEF

La conférence ROADEF¹ est le plus grand événement francophone visant à rassembler des chercheurs de divers domaines, notamment l'optimisation combinatoire, la recherche opérationnelle, la programmation par contraintes et le génie industriel. Cet événement est organisé chaque année et accueille environ 600 participants, avec pour objectif de favoriser les échanges et les collaborations entre chercheurs et professionnels de

l'industrie, ainsi que de contribuer à la formation des jeunes chercheurs et de les encourager à présenter leurs travaux à la communauté nationale. La ROADEF comprend des sessions plénières, des tutoriels en sessions semi-plénières, de multiples sessions parallèles (chacune associée à un ou plusieurs groupes de travail), des sessions de retour d'expériences industrielles et l'assemblée générale de celle-ci. Dans cet article, nous abordons la planification des sessions parallèles de la ROADEF dans les créneaux horaires disponibles tout en évitant les chevauchements entre les groupes de travail. Ce problème se pose chaque année lors de l'organisation de l'événement car chaque session parallèle est généralement prévue pour se dérouler simultanément avec d'autres, ce qui peut potentiellement provoquer des chevauchements et des conflits. En effet, de tels chevauchements empêchent non seulement les chercheurs d'assister aux sessions organisées par leurs groupes de travail respectifs, mais ils peuvent également poser des contraintes concernant les affectations de présidents de session.

Soit S l'ensemble des sessions de la conférence, G l'ensemble des groupes de travail et C l'ensemble des créneaux horaires disponibles. Nous notons que le nombre de papiers attribués à chaque session parallèle peut être différent et nous désignons par L l'ensemble des quantités autorisées de papiers par session parallèle. Nous fixons n le nombre maximum de sessions parallèles pour les créneaux disponibles, qui est déterminé chaque année en fonction des ressources disponibles. Notons qu'en général, une valeur plus élevée pour ce paramètre entraîne une logistique plus complexe et plus de ressources pour gérer les sessions parallèles. Pour chaque session $s \in S$, nous désignons par $np(s)$ et $W(s) \subseteq G$ respectivement le nombre de papiers et l'ensemble des groupes de travail associés à la session s . Enfin, pour chaque créneau $c \in C$, nous désignons par $npMax(c)$ le nombre maximum de papiers autorisés pour chaque session parallèle dans le créneau c .

En 2024, la 25^e édition de ROADEF² a été organisée sur trois jours dans la ville d'Amiens par le laboratoire MIS³ de l'Université de Picardie Jules Verne (UPJV). La conférence comprenait 40 sessions programmées. Elle impliquait également la participation de 20 groupes de travail distincts. Au cours des trois jours de la conférence, il y avait sept créneaux de sessions parallèles. Le comité d'organisation a décidé que chaque session parallèle devait accueillir au moins trois papiers et au maximum six. En effet, étant donné que le temps de présentation et de questions alloué par papier est de 20 minutes, une session avec moins de 3 papiers, c'est-à-dire moins d'une heure, serait trop

1. <https://www.roadev.org/roadev-le-congres-annuel>

2. <https://roadev2024.sciencesconf.org/>

3. <https://mis.u-picardie.fr/>

Slot c	1	2	3	4	5	6	7
npMax(c)	4	6	6	4	4	5	3

TABLE 1 – Nombre maximal de papiers autorisés par créneau

courte tandis qu’une session avec plus de 6 papiers serait trop longue car une pause serait anticipée après 2 heures. Le nombre maximum de sessions parallèles par créneau est fixé à 15 en fonction des ressources disponibles et principalement les salles disponibles sur le site de la conférence. Pour résumer, nous avons $S = \{1, \dots, 40\}$, $G = \{1, \dots, 20\}$, $C = \{1, \dots, 7\}$, $L = \{3, 4, 5, 6\}$ et $n \leq 15$. En outre, nous rapportons le nombre maximum de papiers autorisés pour chaque créneau dans le Tableau 1. Par manque de place, nous ne rapportons pas les valeurs de $np(s)$ et $W(s)$ pour $s \in S$, mais nous mentionnons qu’il y avait un total de 307 papiers acceptés à la conférence répartis sur les différentes sessions. Pour donner un exemple, pour la session $s = 40$, qui est dédiée à la programmation par contraintes, on a $np(s) = 8$ et $W(s) = \{7, 15\}$. Enfin, nous notons que des contraintes supplémentaires pour des sessions spécifiques peuvent survenir. Par exemple, les organisateurs de la session 34 ont demandé à ce que les présentations soient programmées le dernier jour de la conférence, couvert par les créneaux 5, 6 et 7.

4 Modèle Max-SAT

Notre objectif est de minimiser les conflits entre les groupes de travail dans le planning de la conférence, tout en respectant toutes les contraintes liées à l’organisation. Nous commençons d’abord par définir nos variables principales. Plus précisément, nous définissons deux types de variables pour représenter l’allocation des sessions parallèles et pour calculer les conflits comme suit :

- $x_{(s,c,l)}$ est vraie si la session $s \in S$ est affectée au créneau $c \in C$ avec $l \in L$ papiers.
- $y_{(s_1,s_2,c,g)}$ est vraie s’il y a un conflit dans le créneau $c \in C$ pour le groupe de travail $g \in G$ associé à la paire de sessions $(s_1, s_2) \in S^2$ telle que $s_1 < s_2$ et $g \in W(s_1) \cap W(s_2)$

Puisque nous voulons réduire le nombre de conflits de groupe dans la conférence, nous devons clairement minimiser le nombre de variables y satisfaites. Nous pouvons donc définir l’ensemble des clauses souples en conséquence comme suit :

$$\phi_{soft} = \bigwedge_{\substack{(s_1,s_2,c,g) \in S \times S \times C \times G \\ s_1 < s_2 \\ g \in W(s_1) \cap W(s_2)}} \overline{y_{(s_1,s_2,c,g)}}$$

Ensuite, nous énonçons les contraintes qui seront ajoutées à l’ensemble des clauses dures ϕ_{hard} de la formule :

- Au plus une quantité de papiers peut être choisie pour chaque paire (session, créneau) :

$$\sum_{l \in L} x_{(s,c,l)} \leq 1 \quad \forall (s,c) \in S \times C \quad (1)$$

- La subdivision d’une session en créneaux couvre tous les papiers de la session :

$$\sum_{\substack{c \in C \\ l \in L}} x_{(s,c,l)} \times l = np(s) \quad \forall s \in S \quad (2)$$

- L’affectation d’une session à un créneau est valide relativement à la durée maximale du créneau :

$$\bigwedge_{\substack{l \in L \\ l > npMax(c)}} \overline{x_{(s,c,l)}} \quad \forall (s,c) \in S \times C \quad (3)$$

- Le nombre maximal des sessions parallèles par créneau n’est pas dépassé :

$$\sum_{\substack{s \in S \\ l \in L}} x_{(s,c,l)} \leq n \quad \forall c \in C \quad (4)$$

- Deux sessions associées au même groupe et affectées au même créneau génèrent un conflit :

$$(\sum_{l \in L} x_{(s_1,c,l)} \geq 1 \wedge \sum_{l \in L} x_{(s_2,c,l)} \geq 1) \Rightarrow y_{(s_1,s_2,c,g)} \quad (5)$$

$$\forall (s_1, s_2, c, g) \in S^2 \times C \times G, s_1 < s_2 \text{ et } g \in W(s_1) \cap W(s_2)$$

En utilisant les lois de Morgan et la réécriture CNF de l’implication logique, nous pouvons facilement montrer que la contrainte (5) peut être réécrite comme suit :

$$(\bigwedge_{l \in L} \overline{x_{(s_1,c,l)}}) \vee (\bigwedge_{l \in L} \overline{x_{(s_2,c,l)}}) \vee y_{(s_1,s_2,c,g)}$$

Pour réécrire la contrainte en forme CNF, nous pouvons donc appliquer la distributivité de $\wedge$ et $\vee$. Mais, nous proposons d’introduire de nouvelles variables $z_{(s,c)}$ pour chaque paire $(s,c) \in S \times C$ couplées avec l’équivalence logique suivante qui peut être facilement réécrite en forme CNF :

$$z_{(s,c)} \Leftrightarrow \bigwedge_{l \in L} \overline{x_{(s,c,l)}} \quad \forall (s,c) \in S \times C \quad (6)$$

Ainsi, $z_{(s,c)}$ est une nouvelle variable pour chaque paire $(s,c) \in S \times C$ qui sera vraie si la session s n’est pas affectée au créneau c . En utilisant ces variables, nous pouvons maintenant simplement réécrire la contrainte (5) en déclarant que si une paire de sessions $(s_1, s_2) \in S^2$ partage le même groupe et est affectée au même créneau, alors cela implique la variable de conflit correspondante comme suit :

$$z_{(s_1,c)} \vee z_{(s_2,c)} \vee y_{(s_1,s_2,c,g)} \quad (5')$$

$$\forall (s_1, s_2, c, g) \in S^2 \times C \times G, s_1 < s_2 \text{ et } g \in W(s_1) \cap W(s_2)$$

De plus, cette réécriture peut nous permettre de réduire la taille de la contrainte (4) en utilisant $|S|$ variables pour chaque créneau $c \in C$ au lieu de $|S \times L|$ comme suit :

$$\sum_{s \in S} \overline{z_{(s,c)}} \leq n \quad \forall c \in C \quad (4')$$

Enfin, les variables z peuvent également être utilisées pour encoder efficacement des contraintes spécifiques à une session, comme celle qui est survenue pour la session 34 comme suit :

$$\bigwedge_{c \in C \setminus \{5,6,7\}} z_{(34,c)} \quad (7)$$

5 Évaluation expérimentale

Nous avons encodé notre modèle Max-SAT⁴ en utilisant la bibliothèque PySAT⁵ [9]. Nous avons utilisé l'encodage sorting-network [16], qui est disponible pour les contraintes PB et de cardinalité dans PySAT, nous permettant d'encoder efficacement ces contraintes en utilisant $O(m \times \log^2(m))$ nouvelles variables et clauses où $m = \sum_{i=1}^h a_i$ ($m = h$ pour les contraintes de cardinalité). Nous avons généré 7 instances différentes avec les valeurs de n , c'est-à-dire le nombre maximum de sessions parallèles par créneau, variant dans $\{9, \dots, 15\}$. Cela nous permet non seulement minimiser le nombre de conflits de groupes de travail pour une valeur donnée de n mais aussi de rechercher une limite supérieure optimale pour le nombre de sessions parallèles. Étant donné que l'encodage choisi ne dépend pas de cette borne, toutes les instances générées ont 45112 variables et 68062 clauses chacune, comprenant 308 clauses souples.

Nous avons résolu ces instances en utilisant différents solveurs Max-SAT de l'état de l'art : RC2 [10], OpenWBO [15], MaxCDCL [12] et MaxHS [5]. Les tests ont été réalisés sur une machine Dell Inc. Vostro avec un processeur Intel Core i5 cadencé à 3.20GHz, sous Ubuntu 23.10. Un timeout de 3600s a été fixé pour chaque instance. Les résultats en termes de temps de résolution sont décrits dans le Tableau 2. Il est important de mentionner que la solution manuelle proposée par le comité d'organisation comportait 6 conflits avec n fixé à 11. En revanche, notre formulation a permis d'atteindre le coût optimal de 4 conflits pour $n \in \{10, \dots, 15\}$

n	RC2	OpenWBO	MaxCDCL	MaxHS
15	0.779	1.748	0.984	25.606
14	0.897	1.353	0.817	26.035
13	0.805	1.661	1.002	21.867
12	0.949	1.693	0.976	22.090
11	0.843	2.928	1.115	112.050
10	8.927	964.208	173.708	267.181

TABLE 2 – Temps de résolution en secondes

très rapidement. Ainsi, nos résultats ont non seulement réduit le nombre de conflits de groupes de travail, mais montrent également qu'on peut réduire le nombre maximum de sessions parallèles sans impacter la valeur optimale, permettant ainsi d'optimiser les ressources dans la gestion des sessions parallèles. En particulier, le solveur RC2 semble obtenir les meilleurs résultats puisqu'il résout toutes les instances où $n > 10$ en moins de 1s chacune et l'instance avec $n = 10$ en moins de 9s. Le solveur MaxCDCL semble également compétitif avec RC2 pour des valeurs plus élevées de n .

On peut également remarquer que les instances deviennent plus difficiles pour des valeurs plus petites de n . En particulier, le timeout est atteint pour tous les solveurs dans le cas de $n = 9$. En effet, pour cette valeur, l'ensemble des contraintes dures devient insatisfaisable et il semble que les solveurs ont une très grande difficulté à prouver ce résultat. Néanmoins, nous pouvons établir formellement ce résultat en calculant la borne supérieure suivante pour le nombre de papiers : $b = \sum_{c \in C} n \times npMax(c)$. Clairement, pour $n = 9$, nous avons $b = 288 < 307$, et par conséquent il est impossible de planifier toutes les sessions de manière à atteindre le nombre total de papiers.

6 Conclusion

Dans cet article, nous avons abordé un problème d'optimisation dans la planification des sessions de conférences. Plus précisément, nous avons formulé un modèle Max-SAT pour la minimisation des conflits de groupes de travail lors de la conférence ROADEF. Nos résultats indiquent que le nombre de conflits et le nombre maximum de sessions parallèles peuvent être réduits et qu'un planning optimal peut être calculée dans un délai raisonnable. Notre travail est donc très prometteur pour simplifier la tâche de planification, rationaliser la logistique et optimiser les ressources lors des éditions suivantes de la conférence. Comme perspectives, nous prévoyons d'élargir notre recherche en réalisant des expérimentations sur d'autres données, dans le cadre de la ROADEF et au-delà. De plus, nous visons à améliorer notre approche de planification en intégrant des considérations pour les conflits d'auteurs afin de développer un cadre d'optimisation plus complet qui prend en compte les relations complexes entre les participants.

4. Notre code et nos benchmarks sont disponibles sur https://github.com/satoutahhaithem/Scheduling_Problem_JFPC

5. <https://pysathq.github.io/>

Références

- [1] C. ANSÓTEGUI, M. L. BONET et J. LEVY : Sat-based maxsat algorithms. *Artif. Intell.*, 196:77–105, 2013.
- [2] R. J. AS’IN ACH’A et R. NIEUWENHUIS : Curriculum-based course timetabling with SAT and maxsat. *Ann. Oper. Res.*, 218(1):71–91, 2014.
- [3] J. BERG, A. HYTTINEN et M. JÄRVISALO : Applications of maxsat in data analysis. In D. L. BERRE et M. JÄRVISALO, édés : *Proceedings of Pragmatics of SAT 2015, Austin, Texas, USA, September 23, 2015 / Pragmatics of SAT 2018, Oxford, UK, July 7, 2018*, vol. 59 de *EPiC Series in Computing*, p. 50–64. EasyChair, 2018.
- [4] A. BIERE, M. HEULE et H. van MAAREN : *Handbook of Satisfiability : Second Edition*. IOS Press, 2021.
- [5] J. DAVIES et F. BACCHUS : Solving MAXSAT by solving a sequence of simpler SAT instances. In J. H. LEE, éd. : *Principles and Practice of Constraint Programming - CP 2011 - 17th International Conference, Perugia, Italy, September 12-16, 2011. Proceedings*, vol. 6876 de *Lecture Notes in Computer Science*, p. 225–239. Springer, 2011.
- [6] E. DEMIROVIC et N. MUSLIU : Maxsat-based large neighborhood search for high school timetabling. *Comput. Oper. Res.*, 78:172–180, 2017.
- [7] E. DEMIROVIC, N. MUSLIU et F. WINTER : Modeling and solving staff scheduling with partial weighted maxsat. *Ann. Oper. Res.*, 275(1):79–99, 2019.
- [8] E. DEMIROVIĆ, N. MUSLIU et F. WINTER : Modeling and solving staff scheduling with partial weighted maxsat. *Annals of Operations Research*, 275(1):79–99, nov. 2017.
- [9] A. IGNATIEV, A. MORGADO et J. MARQUES-SILVA : Pysat : A python toolkit for prototyping with SAT oracles. In O. BEYERSDORFF et C. M. WINTERSTEIGER, édés : *Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9-12, 2018, Proceedings*, vol. 10929 de *Lecture Notes in Computer Science*, p. 428–437. Springer, 2018.
- [10] A. IGNATIEV, A. MORGADO et J. MARQUES-SILVA : RC2 : an efficient maxsat solver. *J. Satisf. Boolean Model. Comput.*, 11(1):53–64, 2019.
- [11] G. JÄGER, S. CLIMER et W. ZHANG : The complete parsimony haplotype inference problem and algorithms based on integer programming, branch-and-bound and Boolean satisfiability. *J. Discrete Algorithms*, 37:68–83, 2016.
- [12] C. LI, Z. XU, J. COLL, F. MANYÀ, D. HABET et K. HE : Boosting branch-and-bound maxsat solvers with clause learning. *AI Commun.*, 35(2): 131–151, 2022.
- [13] C. M. LI et F. MANYÀ : Maxsat, hard and soft constraints. In A. BIERE, M. HEULE, H. van MAAREN et T. WALSH, édés : *Handbook of Satisfiability - Second Edition*, vol. 336 de *Frontiers in Artificial Intelligence and Applications*, p. 903–927. IOS Press, 2021.
- [14] Y. LI, S. LIN, S. NISHIZAWA et H. ONODERA : Mcell : Multi-row cell layout synthesis with resource constrained MAX-SAT based detailed routing. In *IEEE/ACM International Conference On Computer Aided Design, ICCAD 2020, San Diego, CA, USA, November 2-5, 2020*, p. 157 :1–157 :8. IEEE, 2020.
- [15] R. MARTINS, V. M. MANQUINHO et I. LYNCE : Open-wbo : A modular maxsat solver,. In C. SINZ et U. EGLY, édés : *Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, vol. 8561 de *Lecture Notes in Computer Science*, p. 438–445. Springer, 2014.
- [16] O. ROUSSEL et V. M. MANQUINHO : Pseudo-boolean and cardinality constraints. In A. BIERE, M. HEULE, H. van MAAREN et T. WALSH, édés : *Handbook of Satisfiability - Second Edition*, vol. 336 de *Frontiers in Artificial Intelligence and Applications*, p. 1087–1129. IOS Press, 2021.
- [17] A. SHABANI et B. ALIZADEH : PMTP : A max-sat-based approach to detect hardware trojan using propagation of maximum transition probability. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 39(1):25–33, 2020.