

Minimisation des pics d'énergie dans l'équilibrage des lignes d'assemblage par la Satisfiabilité Maximum

Zhifei Zheng ^{2*}Sami Cherif ¹Rui Sá Shibasaki ¹¹ Laboratoire MIS UR 4290, Université de Picardie Jules Verne, Amiens, France² Engineering Research Center of Cyberspace & School of Software, Yunnan University, China
zzfrey@mail.ynu.edu.cn {sami.cherif;rui.sa.shibasaki}@u-picardie.fr

Résumé

Le problème SALB3PM (Simple Assembly Line Balancing Problem with Power Peak Minimization) est un problème relativement nouveau qui vise à assigner des tâches à des machines tout en visant la minimisation des pics d'énergie consommée par l'ordonnancement obtenu. Cet article présente des nouveaux modèles pour SALB3PM basés sur la Satisfiabilité Maximum (MaxSAT), offrant ainsi une nouvelle perspective pour résoudre ce problème de manière efficace. Les résultats expérimentaux démontrent l'intérêt de notre approche. Notamment, à notre connaissance, notre travail représente une tentative inaugurale visant à résoudre le problème SALB3PM à travers MaxSAT.

Abstract

The Simple Assembly Line Balancing Problem with Power Peak Minimization (SALB3PM) is a relatively new problem that aims to assign tasks to workstations with a focus on minimizing power peaks. This paper introduces novel models for SALB3PM based on Maximum Satisfiability (MaxSAT), the natural optimization extension of the Satisfiability problem, providing a new perspective to solve this optimization problem effectively. Experimental results demonstrate the efficiency and robustness of our approach. Notably, to the best of our knowledge, our work represents an inaugural attempt to address the SALB3PM problem through the lens of Maximum Satisfiability.

1 Introduction

Le problème SALB3PM (Simple Assembly Line Balancing Problem with Power Peak Minimization), initialement introduit dans [5], consiste à trouver une affectation réalisable de tâches à des stations de travail qui minimise le pic global de consommation d'énergie

tout en respectant les contraintes de précédence et le temps de cycle. Étant la première variante de SALBP qui prend en compte les pics d'énergie à un niveau stratégique, ce problème a attiré l'attention des chercheurs, conduisant à la proposition de deux variantes récentes : SALB3PM-ESD [4] et PSCALB3PM [3]. Sur le plan pratique, un modèle ILP a été proposé dans [5] et une métaheuristique a été développée dans [7]. Récemment, une nouvelle méthode a été introduite dans [10], basée sur le formalisme de Satisfiabilité (SAT) pour résoudre SALBP3PM efficacement, donnant des résultats prometteurs.

Dans cet article, nous proposons un nouveau modèle de SALB3PM basé sur la Satisfiabilité Maximum (Max-SAT), en nous appuyant sur le travail fondateur de Gianessi et al. [5] et les contributions ultérieures de Py et Tuyaba [10]. Notre modèle vise à exploiter la puissance de MaxSAT dans la résolution de problèmes d'optimisation complexes dans les domaines de l'ordonnancement et de la planification. Cet article est organisé comme suit. La Section 2 comprend les définitions et notations nécessaires. Nos modèles Max-SAT dédiés à SALBP3PM sont introduits dans la Section 3 et évalués dans la Section 4. En fin, nous concluons et discutons des travaux futurs dans la Section 5.

2 Préliminaires

2.1 SALBP3PM

Soit N et S respectivement un ensemble de n tâches et m stations de travail, pour chaque $i \in N$, la tâche i a un temps d'exécution $t_i \in \mathbb{N}^*$ et une consommation d'énergie $w_i \in \mathbb{N}^*$. Nous notons c le temps de cycle après lequel toutes les stations de travail doivent être finies. Ce temps est divisé en unités comme suit $T =$

*Papier étudiant : Zhifei Zheng ² est auteur principal.

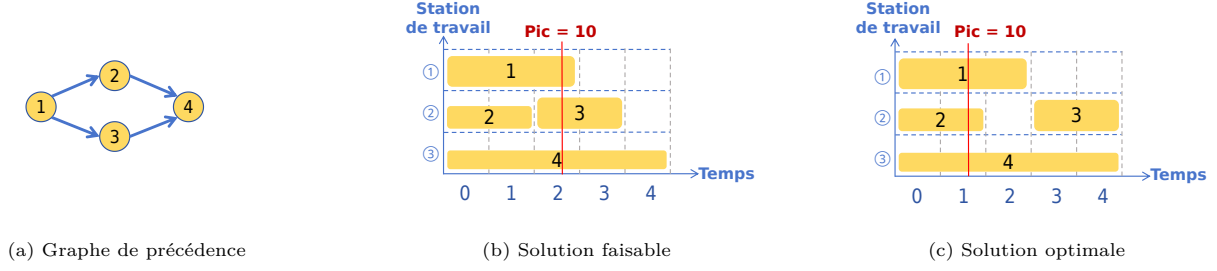


FIGURE 1 – Instance SALB3PM avec $n = 4$, $m = 3$ et $c = 5$

[1, c]. Chaque poste de travail peut gérer au plus une tâche à la fois.

On note P l'ensemble des relations de précedence entre les tâches. Si $(i, j) \in P$, noté $i \prec j$, alors soit la tâche i est affectée à une station de travail avec un indice plus petit que celui de la tâche j , soit elles sont assignées à la même station de travail avec i programmée pour commencer plus tôt. Le problème SALB3PM consiste donc à affecter des tâches aux stations de travail de manière à minimiser le pic de consommation énergétique tout en respectant les contraintes de précedence et le temps de cycle.

La Figure 1 illustre deux affectations réalisables pour une instance SALB3PM avec 4 tâches, 3 stations de travail, un temps de cycle de 5, les durées des tâches $(t_i)_{1 \leq i \leq 4} = (3, 2, 2, 5)$, les consommations énergétiques $(W_i)_{1 \leq i \leq 4} = (4, 3, 4, 2)$, et les contraintes de précedence $P = \{(1, 2), (1, 3), (2, 4), (3, 4)\}$. La première affectation (Figure 1b) conduit à une pic de consommation énergétique maximal de 10, tandis que la deuxième affectation (Figure 1c) résulte en un pic de 9 et correspond à la solution optimale pour cette instance.

2.2 Satisfiabilité Maximum

Soit V un 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 (FNC) est une conjonction de clauses. Une affectation $\alpha : V \rightarrow \{true, false\}$ associe à chaque variable une valeur booléenne et peut être représentée comme un ensemble de littéraux. Un littéral l est satisfait par une affectation α si $l \in \alpha$, sinon il est falsifié par α . Une clause est satisfaite par une affectation α si au moins un de ses littéraux est satisfait par α , sinon elle est falsifiée par α . Une formule FNC est satisfiable s'il existe une affectation α qui satisfait toutes ses clauses, sinon elle est dite insatisfiable. Le coût d'une affectation α , noté $cost_\alpha(\phi)$, est le nombre de clauses de la formule ϕ falsifiées par α .

Résoudre le problème de Satisfiabilité (SAT) consiste à déterminer si une formule FNC donnée ϕ est satisfiable, tandis que pour la Satisfiabilité Maximum (MaxSAT), l'objectif est de déterminer le nombre maximum de clauses dans ϕ pouvant être satisfaites par une af-

fectation des variables ou, de manière équivalente, le nombre minimum de clauses que chaque affectation doit falsifier, c'est-à-dire $opt(\phi) = \min_{\alpha} cost_\alpha(\phi)$ [8]. Le problème de Max-SAT partiel est une variante bien connue prenant en entrée une formule FNC 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 est donc de calculer $opt(S)$ tel que toutes les clauses de H soient satisfaites. Une autre variante bien connue est Max-SAT pondéré, qui prend en entrée une formule FNC pondérée ϕ , c'est-à-dire un ensemble de tuples (C, w_C) où C est une clause et $w_C \in \mathbb{N}^*$ est son poids associé (les clauses dures sont typiquement associées à un poids infini). Dans ce cas, on cherche $opt(\phi)$ où $cost_\alpha(\phi)$ désigne désormais la somme des poids des clauses de ϕ falsifiées par α . Pour fin, nous mentionnons que la littérature antérieure introduit des contraintes pseudo-booléennes dans le contexte du SAT et de Max-SAT [2]. Ces contraintes sont de la forme $\sum_j a_j l_j \triangleright b$ où $a_j, b \in \mathbb{N}$, l_j est un littéral et $\triangleright \in \{=, >, \geq, <, \leq\}$.

3 Modèles MaxSAT pour SALB3PM

3.1 Modèle initial

Une instance SALB3PM peut être naturellement transformée en une instance MaxSAT comme suit. Nous notons $T^i = \{1, \dots, c - t_i + 1\}$ l'ensemble des temps de début d'exécution possibles pour la tâche i . Pour trouver la valeur optimale du pic de consommation énergétique W_{peak} , nous établissons une borne inférieure et une borne supérieure comme suit : $LB = \max_{1 \leq i \leq n} w_i$ et $UB = \sum_{w_i \in W_m} w_i$ où W_m est l'ensemble des m valeurs les plus élevées de w_i . Ces bornes nous permettent de réduire l'espace de recherche, en se concentrant uniquement sur les zones contenant la solution optimale.

On définit les variables booléennes suivantes pour notre modèle :

- **Variables d'affectation** $X_{i,s}$ avec $i \in N$ et $s \in S$, qui sont égales à 1 si la tâche i est affectée à la station s , 0 sinon.
- **Variables de déclenchement** $B_{i,t}$ avec $i \in N$

et $t \in T$, qui sont égales à 1 lorsque la tâche i commence à l'unité de temps t , 0 sinon.

- **Variables d'activité** $A_{i,t}$ avec $i \in N$ et $t \in T$, qui sont égales à 1 lorsque la tâche i est exécutée à l'unité de temps t , 0 sinon.
- **Variables de solution** U_j avec $j \in [1, UB]$, qui sont égales à 1 lorsque la consommation de puissance maximale obtenue est inférieure ou égale à j , 0 sinon. Ainsi, on a $W_{peak} = \sum_{j \in [1, UB]} U_j$.

La Figure 2 montre notre modèle MaxSAT initial avec les contraintes 1 à 11. Ce modèle minimise le pic de consommation énergétique en utilisant des clauses souples dans la contrainte 1 impliquant des littéraux $\overline{U_j}$ pour chaque unité d'énergie. Les clauses dures imposent les contraintes de précédence et de temps de cycle. La contrainte 2 garantit que chaque tâche est affectée à exactement une station. Les contraintes 3 et 4 sont nécessaires pour satisfaire les relations de précédence, assurant ainsi que les tâches sont effectuées dans le bon ordre. La contrainte 5 garantit que chaque tâche débute à exactement une unité de temps faisable. La contrainte 6 impose que les tâches doivent commencer dans leurs fenêtres de temps faisables. Pour empêcher l'exécution simultanée de tâches sur la même station, les contraintes 7 et 8 sont utilisées, assurant une gestion adéquate de l'activation des tâches et des conflits d'exécution. Les contraintes 9, 10 et 11 sont liées aux

FIGURE 2 – Modèle MaxSAT initial pour SALB3PM

$$\text{Clauses souples : } \overline{U_j} \quad \forall j \in [1, UB] \quad (1)$$

$$\text{Clauses dures : } \sum_{s \in S} X_{i,s} = 1 \quad \forall i \in N \quad (2)$$

$$\overline{X_{i,s_1}} \vee \overline{X_{j,s_2}} \quad \forall (i, j, s_1, s_2) \in N^2 \times S^2 \text{ s.t. } i \prec j \text{ and } s_1 > s_2 \quad (3)$$

$$\overline{X_{i,s}} \vee \overline{X_{j,s}} \vee \overline{B_{i,t_1}} \vee \overline{B_{j,t_2}} \quad \forall (i, j, s, t_1, t_2) \in N^2 \times S \times T^2 \text{ s.t. } i \prec j \text{ and } t_1 > t_2 \quad (4)$$

$$\sum_{t \in T} B_{i,t} = 1 \quad \forall i \in N \quad (5)$$

$$\overline{B_{i,t}} \quad \forall (i, t) \in N \times T \setminus T^i \quad (6)$$

$$\overline{B_{i,t}} \vee A_{i,t+\epsilon} \quad \forall (i, t, \epsilon) \in N \times T^i \times [0, t_i - 1] \quad (7)$$

$$\overline{X_{i,s}} \vee \overline{X_{j,s}} \vee \overline{A_{i,t}} \vee \overline{A_{j,t}} \quad \forall (i, j, s, t) \in N^2 \times S \times T \text{ s.t. } i < j \quad (8)$$

$$U_j \quad \forall j \in [1, LB] \quad (9)$$

$$\overline{U_j} \vee U_{j-1} \quad \forall j \in [2, UB] \quad (10)$$

$$\sum_{i \in [1, n]} w_i * A_{i,t} + \sum_{j \in [1, UB]} \overline{U_j} \leq UB \quad \forall t \in T \quad (11)$$

pics d'énergie. En particulier, la contrainte 9 impose la borne inférieure calculée. La contrainte 10 maintient l'ordre d'utilisation des unités d'énergie. Enfin, la contrainte 11 encapsule la contrainte de consommation d'énergie globale, en s'assurant qu'elle reste dans UB pour toutes les unités de temps.

3.2 Modèle avec représentation binaire

Dans MaxSAT, l'espace de recherche potentiel est limité par le nombre de variables booléennes. Étant donné que UB peut avoir des valeurs élevées, nous proposons d'introduire un modèle basé sur une représentation binaire pour réduire le nombre de variables. Plus précisément, nous remplaçons les variables de solution présentées à la Section 3.1 par des variables $binU_j$ avec $j \in [0, \lceil \log_2 UB \rceil]$. Ainsi, $binU_j$ prend la valeur 1 lorsque le j -ème bit de la solution obtenue est 1 et 0 sinon. Notons que, dans ce cas, on a $W_{peak} = \sum_{j \in [0, \lceil \log_2 UB \rceil]} 2^j \cdot binU_j$.

Par exemple, si la solution est $6 = (110)_2$, alors les variables de solution encodées en binaire seraient $binU_2 = 1$, $binU_1 = 1$, et $binU_0 = 0$. Dans le nouveau modèle, on fixe également $UB' = \sum_{j \in [0, \lceil \log_2 UB \rceil]} 2^j$. La représentation binaire peut alors être intégrée dans le modèle initial en gardant les contraintes 3 à 8 et en implémentant des modifications spécifiques pour les autres contraintes comme illustré dans la figure 3. En particulier, la contrainte 1' vise à minimiser le pic maximal d'énergie en utilisant une représentation binaire, où chaque unité binaire $binU_j$ est pondérée par 2^j . Cette approche permet une représentation plus efficace des valeurs des pics de puissance. Les clauses dures du modèle initial issues des contraintes 2 à 8 sont conservées pour imposer les contraintes d'affectation des tâches, des relations de précédence et de la planification temporelle. La contrainte 9' est une adaptation de l'équation 9, qui garantit que la valeur finale du pic obtenue, représentée au format binaire, est supérieure ou égale à la borne inférieure (LB). La contrainte 11' est

FIGURE 3 – Modèle MaxSAT avec représentation binaire

Clauses souples :

$$(\overline{binU_j}, 2^j) \quad \forall j \in [0, \lceil \log_2 UB \rceil] \quad (1')$$

Clauses dures :

$$\text{Contraintes 2 à 8 du modèle initial maintenues} \quad (2-8)$$

$$\sum_{j \in [0, \lceil \log_2 UB \rceil]} 2^j binU_j \geq LB \quad \forall t \in T \quad (9')$$

$$\sum_{i \in [1, n]} w_i * A_{i,t} + \sum_{j \in [0, \lceil \log_2 UB \rceil]} 2^j * \overline{binU_j} \leq UB' \quad \forall t \in T \quad (11')$$

une adaptation de l'équation 11, utilisant des variables binaires, assurant que la puissance totale utilisée à tout moment reste dans les limites du pic d'énergie maximal calculé. Ensemble, les nouvelles contraintes, aux côtés de celles maintenues, forment le nouveau modèle MaxSAT avec représentation binaire pour SALB3PM, offrant un espace de solution plus compact comparé au modèle initial.

4 Évaluation expérimentale

Les expérimentations ont été réalisées sur une machine HP OMEN sous Ubuntu 22.04.4 LTS et équipée d'un processeur Intel Core i9-13900HX de 13^{ème} génération (5,40GHz). Pour nos tests, nous avons utilisé un ensemble d'instances adaptés à partir de SALBP-1¹, comme dans [5] et [10]. Nous avons employé trois solveurs de l'état de l'art : CPLEX² pour la méthode ILP proposée dans [5], Cadical [1] pour la méthode SAT proposée dans [10], et MaxHS [9] pour résoudre nos instances MaxSAT. Pour effectuer l'encodage des instances en FNC, nous avons utilisé PySAT [6], avec les contraintes pseudo-boléenne encodées en mode "Best", signifiant que PySAT détermine automatiquement le meilleur encodage parmi ceux disponible dans

1. <https://assembly-line-balancing.de/salbp/benchmark-data-sets-1993/>
2. <https://www.ibm.com/fr-fr/products/ilog-cplex-optimization-studio>

TABLE 1 – Comparaison expérimentale des méthodes ILP, SAT et MaxSAT. "-" est reporté pour les instances non résolues à l'optimum dans le temps limite pourvu.

Benchmark	n	m	c	ILP	SAT	MaxSAT (<i>MaxHS</i>)	
				<i>CPLEX</i>	<i>Cadical</i>	Initial	Binary
MERTENS-1	7	6	6	0.60	0.005	1.09	0.01
MERTENS-2	7	2	18	122.25	0.06	0.45	0.035
BOWMAN8-1	8	5	20	0.23	0.04	1.63	0.032
JAESCHKE-1	9	8	6	0.03	0.006	1.21	0.01
JAESCHKE-2	9	3	18	1.69	0.12	1.47	0.096
JACKSON-1	11	8	7	0.06	0.021	2.96	0.03
JACKSON-2	11	3	21	20.21	0.46	3.29	0.121
MANSOOR-1	11	4	48	9.03	0.181	10.46	0.28
MANSOOR-2	11	2	94	415.00	0.942	5.02	1.47
MITCHELL-1	21	8	14	17.31	0.289	26.56	0.50
MITCHELL-2	21	3	39	448.46	4.195	20.29	5.67
ROSZIEG-1	25	10	14	398.53	6.56	228.46	3.666
ROSZIEG-2	25	4	32	1192.62	4.405	29.89	7.90
HESKIA-1	28	8	138	-	-	-	68.982
HESKIA-2	28	3	342	-	-	-	-
BUXEY-1	29	13	27	-	-	1090.13	77.141
BUXEY-2	29	7	54	-	-	-	-
SAWYER30-1	30	14	25	78.99	0.902	182.72	4.69
SAWYER30-2	30	5	75	-	-	-	-
GUNTHER-1	35	14	41	-	4.046	-	7.83
GUNTHER-2	35	7	81	-	-	-	-

la librairie. Les temps d'exécution (en secondes) pour chaque méthode sont reportés dans le tableau 1. Pour chaque instance, le nombre de tâches (n), le nombre de stations (m) et le temps de cycle (c) sont indiqués. D'après les résultats expérimentaux, nous observons que le modèle MaxSAT avec la représentation binaire obtient les meilleurs résultats, avec 17 instances résolues sur 21, surpassant les modèles ILP (resp. 14) et SAT (resp. 15).

5 Conclusion

Ce papier propose des modèles MaxSAT dédiés au problème SALB3PM. En particulier, un modèle MaxSAT avec une représentation binaire des variables de solution est introduit, permettant de réduire considérablement l'espace de recherche. Les résultats expérimentaux montrent l'efficacité de notre modèle. À notre connaissance, ce travail représente la première tentative visant à aborder le problème SALB3PM à travers le prisme de la Satisfiabilité Maximum. Dans les travaux futurs, nous élargirons nos expérimentations à un ensemble de données plus grand et à plus de solveurs tout en améliorant notre modèle. Notre approche a également le potentiel d'être appliquée à d'autres variantes de problème d'équilibrage de lignes d'assemblage.

Références

- [1] A. BIERE, K. FAZEKAS, M. FLEURY et M. HEISINGER : CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020. *In Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, p. 51–53. University of Helsinki, 2020.
- [2] A. BIERE, M. HEULE et H. van MAAREN : *Handbook of satisfiability*, vol. 185. IOS press, 2009.
- [3] X. DELORME et P. GIANESSI : Designing reconfigurable manufacturing systems to minimize power peak. *IFAC-PapersOnLine*, 55(10):1296–1301, 2022.
- [4] X. DELORME, P. GIANESSI et D. LAMY : A new decoder for permutation-based heuristics to minimize power peak in the assembly line balancing. *IFAC-PapersOnLine*, 56(2):3704–3709, 2023.
- [5] P. GIANESSI, X. DELORME et O. MASMOUDI : Simple assembly line balancing problem with power peak minimization. *In IFIP APMS*, p. 239–247. Springer, 2019.
- [6] A. IGNATIEV, A. MORGADO et J. MARQUES-SILVA : PySAT : A Python toolkit for prototyping with SAT oracles. *In SAT*, p. 428–437, 2018.
- [7] D. LAMY, X. DELORME et P. GIANESSI : Line balancing and sequencing for peak power minimization. *IFAC-PapersOnLine*, 53(2):10411–10416, 2020.
- [8] C. M. LI et F. MANYA : Maxsat, hard and soft constraints. *In Handbook of satisfiability*, p. 903–927. IOS Press, 2021.
- [9] A. NISKANEN, J. BERG et M. JÄRVISALO : Maxhs in maxsat evaluation 2022. *MaxSAT Evaluation 2022*, p. 35, 2022.
- [10] M. PY et A. TUYABA : Utilisation de sat pour résoudre le problème salbp avec minimisation du pic de consommation. *In JFPC*, 2023.