



**HAL**  
open science

## Approche portfolio pour le problème des p-centres

Thomas Bazaille, Chumin Li, Corinne Lucet, Laure Brisoux Devendeville,  
Sami Cherif

► **To cite this version:**

Thomas Bazaille, Chumin Li, Corinne Lucet, Laure Brisoux Devendeville, Sami Cherif. Approche portfolio pour le problème des p-centres. Journées Francophones de Programmation par Contraintes (JFPC 2024), Jun 2024, Lens, France. hal-04701887

**HAL Id: hal-04701887**

**<https://hal.science/hal-04701887v1>**

Submitted on 18 Sep 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Approche portfolio pour le problème des p-centres

Thomas Bazaille\*

Chumin Li

Corinne Lucet

Laure Brisoux Devendeville

Sami Cherif

Laboratoire MIS (UR 4290), Université de Picardie Jules Verne, France

{thomas.bazaille,chumin.li,corinne.lucet,laure.devendeville,sami.cherif}@u-picardie.fr

## Résumé

Le problème des p-centres relève de la catégorie des problèmes de localisation et est particulièrement utile pour déterminer l'emplacement de ressources au sein d'un réseau. Dans cet article, nous introduisons une nouvelle méthode pour résoudre le problème des p-centres que nous allons combiner avec un algorithme existant pour former une approche portfolio. La méthode proposée est basée sur une encodage du problème des p-centres par le biais de la Satisfiabilité Maximale (MaxSAT). Notre formulation est comparée au modèle linéaire binaire de [3], implémenté sous CPLEX. Pour tirer le meilleur parti de chaque méthode, nous proposons une approche portfolio qui donne des résultats meilleurs sur les instances de la bibliothèque TSPLIB, avec 3% d'instances résolues en plus en utilisant une combinaison de deux solveurs de l'état de l'art, à savoir MaxCDCL et CPLEX.

## Abstract

The p-center problem falls under the category of location problems, and is particularly useful for determining the placement of resource centers within a network. In this paper, we introduce a new method for solving the vertex restricted p-center problem and we aim to combine it with an existing algorithm to form a portfolio. The developed method is based on an encoding of the p-center problem through Maximum Satisfiability (MaxSAT), the natural optimization extension of the Satisfiability problem. Our method is compared to the binary linear model of [3], implemented under CPLEX. To get the best of each model, we propose a portfolio approach which gives better results on instances from the TSPLIB library as we manage to solve 3% more instances optimally by using a combination of two state-of-the-art optimization solvers, namely MaxCDCL and CPLEX.

## 1 Introduction

Le problème des p-centres appartient à la famille des problèmes de localisation d'installations et consiste à minimiser la distance de service maximale entre les installations candidates et les nœuds de demande [2]. Les applications de ce problème sont nombreuses, telles que l'emplacement des casernes de pompiers, des dépôts d'ambulances, des stations de recharge pour véhicules électriques, etc. Les problèmes de localisation d'installations peuvent être classés en types continus et discrets. Dans la version continue, les installations peuvent être placées n'importe où dans une région continue tandis que, dans la version discrète, les installations sont limitées à un ensemble fini de sites candidats, réduisant la charge de traitement des données et la dimension du problème. La majorité des problèmes de localisation dans la littérature relève de la catégorie discrète, et notre cas ne fait pas exception puisque le problème des p-centres est NP-difficile même pour la version discrète [9].

Le concept d'optimisation du placement des centres d'installation dans un graphe a été initialement formalisé par Hakimi [7], motivé par des applications dans les télécommunications et le déploiement stratégique des policiers au niveau des réseaux autoroutiers. Depuis le travail pionnier de Hakimi, plusieurs heuristiques et méthodes exactes ont été proposées pour résoudre ce problème [6, 4, 8, 5, 3, 2, 13]. En particulier, dans [14], il a été montré que le problème des p-centres pouvait être interprété comme une séquence de problèmes de couverture de sommets dans un graphe (Dominating Set). Dans cet article, nous proposerons une approche reposant sur la résolution d'une série de problèmes de couverture via la satisfiabilité maximale (MaxSAT), l'extension naturelle d'optimisation du problème SAT, qui consiste à trouver le nombre maximal de contraintes

\*Papier doctorant : Thomas Bazaille est auteur principal.

clausales pouvant être satisfaites par une interprétation des variables dans une formule donnée. Nous résolvons donc des instances des p-centres en utilisant des appels itératifs à des solveurs MaxSAT de l'état de l'art et nous intégrons cette méthode dans une approche portfolio, montrant que nous pouvons obtenir des résultats légèrement meilleurs sur des instances de la bibliothèque TSPLIB.

Cet article est organisé comme suit. Dans la Section 2, nous fournissons les définitions et notions nécessaires sur le problème des p-centres et MaxSAT puis nous passons en revue les travaux connexes. Dans la Section 3, nous proposons une formulation itérative grâce à MaxSAT pour le problème des p-centres et nous présentons notre stratégie portfolio. Dans la Section 4, nous rapportons des résultats des différentes méthodologies. Enfin, nous concluons et discutons des travaux futurs dans la Section 5.

## 2 Préliminaires

### 2.1 Problème des p-centres

Soit  $G = (S, E)$  un graphe non orienté, où  $S$  est l'ensemble de sommets,  $E$  l'ensemble des arêtes, et soit  $p$  le nombre de centres à placer. La distance du plus court chemin entre les sommets  $u$  et  $v$  est notée  $\delta(u, v)$ . Le problème des p-centres est un problème de localisation d'installations qui consiste à trouver un ensemble de centres  $V \subseteq S$ , tel que  $|V| \leq p$  en minimisant la fonction objectif suivante :

$$f = \max_{v \in S} \min_{u \in V} \delta(u, v)$$

Depuis les travaux pionniers de Hakimi [7], plusieurs formulations en programmation linéaire du problème des p-centres avec des variables binaires ont été introduites. La première a été présentée par Daskin dans [4]. Plus tard, Calik a proposé une amélioration du modèle de Daskin dans [3], permettant de résoudre le problème comme une succession de problèmes de décision. Inspirés par les travaux de Minieka [14], Ilhan et Pinar ont été les premiers à proposer de résoudre le problème des p-centres comme une succession de problèmes de Dominating Set (DS) dans [8]. Pour une distance donnée  $d \in \mathbb{N}$ , nous définissons le graphe  $G_d = (S, E_d)$  induit par  $G$  avec  $E_d = \{(v, v') \in S \times S \mid \delta(v, v') \leq d\}$ . Le problème DS consiste ainsi à rechercher un sous-ensemble  $V \subset S$  de centres tel que  $|V| \leq p$  et pour tout  $v \in S, \exists v' \in V$  tel que  $(v, v') \in E_d$ . Ces sous-problèmes de décision peuvent être formulés par le programme linéaire suivant :

$$\sum_{v' \in N_{G_d}(v)} x_{v'} \geq 1, \quad \forall v \in S \quad (1)$$

$$\sum_{v \in S} x_v \leq p \quad (2)$$

$$x_v \in \{0, 1\}, \quad \forall v \in S \quad (3)$$

```

min ← 1 ; // indice de la première distance
max ← M ; // indice de la dernière distance
while max - min > 1 do
    h ← (max+min)/2 ;
    if DS(d_h) then
        | max ← h ;
    else
        | min ← h + 1 ;
    end
end

```

**Algorithme 1** : Résolution des p-centres par une succession d'ensembles dominants [8]

où  $x_v$  est une variable binaire qui sera égale à 1 si un centre est placé sur le sommet  $v$  et à 0 sinon, et  $N_{G_d}(v)$  est l'ensemble de tous les sommets connectés à  $v$  dans  $G_d$ . Notons que, si le problème DS est faisable pour une certaine distance  $d \in \mathbb{N}$ , alors le problème des p-centres a nécessairement une solution dont le coût est inférieur ou égal à  $d$ . Inversement, si le problème DS n'admet pas de solution, alors le coût optimal pour le problème des p-centres est supérieur à  $d$ . Nous pouvons ainsi procéder par dichotomie sur l'ensemble des distances faisables ordonnées  $d_1 < d_2 < \dots < d_M$ , en résolvant une itération du problème DS pour chacune d'elles afin de résoudre le problème des p-centres comme illustré dans l'algorithme 1. Plus récemment, Liu et al. ont également appliqué une approche DS pour résoudre le problème des p-centres mais basée sur la satisfiabilité propositionnelle (SAT) comme paradigme de décision inhérent [13].

### 2.2 Satisfiabilité maximum

Soit  $V$  l'ensemble des variables propositionnelles. Un littéral  $l$  est une variable  $x \in X$  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  $\alpha : V \rightarrow \{\text{vrai}, \text{faux}\}$  associe à chaque variable une valeur booléenne et peut être représentée comme un ensemble de littéraux. Un littéral  $l$  est satisfait par une interprétation  $\alpha$  si  $l \in \alpha$ , sinon il est falsifié par  $\alpha$ . Une clause est satisfaite par une interprétation  $\alpha$  si au moins un de ses littéraux est satisfait par  $\alpha$ , sinon elle est falsifiée par  $\alpha$ . Le coût d'une interprétation  $\alpha$ , noté  $cost_\alpha(\phi)$ , est le nombre de clauses dans la formule  $\phi$  falsifiées par  $\alpha$ .

Résoudre le problème de satisfiabilité maximum (MaxSAT) consiste à déterminer le nombre maximum de clauses qui peuvent être satisfaites par une interprétation des variables dans une formule CNF donnée  $\phi$  ou, de manière équivalente, le nombre minimum de clauses que chaque interprétation doit falsifier, c'est-

à-dire  $opt(\phi) = \min_{\alpha} cost_{\alpha}(\phi)$ . Le problème MaxSAT partiel est une variante bien connue qui prend en entrée une formule CNF partielle, c'est-à-dire un ensemble bipartite 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 le MaxSAT classique. MaxSAT a de nombreuses applications dans des domaines variés tels que l'intelligence artificielle, la vérification formelle ainsi que la conception de matériels et de logiciels, ce qui en fait un formalisme significatif et puissant dans le domaine de l'optimisation combinatoire [12].

### 3 Approche portfolio pour p-centre

Afin d'introduire notre approche portfolio, nous commençons tout d'abord par un encodage du problème DS pour une distance donnée  $d$  en MaxSAT. Pour cela, nous devons vérifier que chaque sommet a au moins un centre dans son voisinage ou qu'il est lui-même un centre dans  $G_d$ . Tout d'abord, nous définissons la clause  $C_v$  qui impose que le sommet  $v \in S$  soit dominé par au moins un centre comme suit :

$$C_v = \bigvee_{v' \in N_{G_d}(v)} x_{v'}$$

Ainsi, la formule CNF suivante impose les règles de dominance, assurant que chaque sommet du graphe est dominé.

$$\Phi_{hard} = \bigwedge_{v \in S} C_v$$

La formule ci-dessus représente donc l'ensemble des clauses dures qui doivent être satisfaites pour résoudre le problème DS. De plus, nous devons limiter le nombre de centres afin de trouver une solution au problème contenant moins de  $p$  centres. À cette fin, nous ajoutons  $|S|$  clauses souples comme suit :

$$\phi_{soft} = \bigwedge_{v \in S} \neg x_v$$

Ces clauses unitaires limiteront le nombre de centres, car l'objectif du MaxSAT est de maximiser le nombre de clauses souples satisfaites. Enfin, la formule  $\phi = \phi_{soft} \cup \phi_{hard}$  encode le problème de l'ensemble dominant minimal et peut être traitée par des solveurs dédiés au problème MaxSAT partiel.

Dans la mise en œuvre de l'algorithme 1, une méthode différente peut être employée pour traiter chaque problème de l'ensemble dominant à chaque itération. Cette approche repose sur le fait que toutes les approches ne résolvent pas de manière uniforme toutes les instances. Par conséquent, notre objectif est d'amalguer diverses méthodes et de tirer profit de la puissance

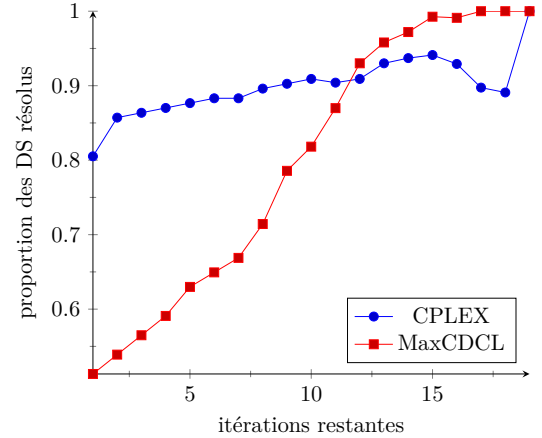


FIGURE 1 – Proportion de problèmes DS résolus en fonction du nombre d'itérations restantes dans l'algorithme 1

de chacune pour une résolution efficace du problème DS et donc des p-centres. Ce faisant, nous visons à établir des règles de décision guidant la sélection d'un solveur spécifique en fonction de ses performances pour une instance donnée. Nous maintenons donc l'opération de l'algorithme 1 mais avec des formulations différentes possibles pour DS à chaque itération, la première utilisant la formulation linéaire décrite dans la section 2.1 et la seconde basée sur notre modèle MaxSAT décrit ci-dessus. Comme un solveur MaxSAT cherche à minimiser le nombre de centres utilisés pour dominer les sommets d'un graphe donné  $G_d$ , nous pouvons adapter le code du solveur pour qu'il arrête son calcul si une solution contenant moins de  $p$  centres a été trouvée.

### 4 Évaluation expérimentale

Nous avons présenté deux méthodes pour résoudre le problème de l'ensemble dominant, la première utilisant une formulation en programme linéaire (LP) et la seconde utilisant une formulation MaxSAT. Pour résoudre le problème linéaire, nous utilisons le solveur CPLEX, et pour le problème MaxSAT, nous utilisons le solveur MaxCDCL [10, 11]. Nous avons implémenté l'algorithme 1 où le problème de l'ensemble dominant peut être résolu en utilisant l'une des deux approches avec le solveur correspondant à chaque itération. Les calculs ont été effectués sur la plateforme MatriCS<sup>1</sup>, sur des serveurs Intel(R) Xeon(R) CPU E5-2680 cadencés à 2,40GHz avec 128 Go de RAM. Les tests ont été réalisés sur un ensemble de benchmarks comprenant 231 instances, extraites de 21 instances de la bibliothèque TSPLIB<sup>2</sup> en définissant 11 valeurs différentes de  $p$  dans

1. <https://www.matrics.u-picardie.fr/>  
2. <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>

|              | MaxSAT     |                 | LP         |                 | LP+MaxSAT  |                  |
|--------------|------------|-----------------|------------|-----------------|------------|------------------|
| d1291        | 8          | 2158,585        | 11         | <b>1996,012</b> | 11         | 2022,741         |
| d1655        | 5          | 4110,968        | 11         | <b>3843,687</b> | 11         | 3927,293         |
| d657         | 8          | 133,3           | 11         | <b>3307,229</b> | 11         | 3473,972         |
| f13795       | 8          | 1564,809        | 9          | 2029,095        | 11         | <b>2884,011</b>  |
| f1417        | <b>11</b>  | <b>18,405</b>   | 11         | 112,194         | 11         | 98,786           |
| fnl4461      | <b>1</b>   | <b>245,664</b>  | 0          | -               | 0          | -                |
| pcb3038      | 1          | 54,457          | 0          | -               | <b>2</b>   | <b>4554,674</b>  |
| pr1002       | 7          | 1432,252        | <b>11</b>  | <b>2547,607</b> | 11         | 2548,296         |
| pr136        | 11         | 10,917          | <b>11</b>  | <b>228,891</b>  | 11         | 238,036          |
| pr226        | 11         | 70,92           | 11         | 235,43          | <b>11</b>  | <b>20,096</b>    |
| rl1304       | 7          | 2539,704        | 11         | 3368,421        | <b>11</b>  | <b>1938,975</b>  |
| rl1323       | 6          | 367,59          | <b>11</b>  | <b>5759,898</b> | 11         | 6424,659         |
| rl1889       | 5          | 118,125         | 5          | 1246,706        | <b>7</b>   | <b>5940,424</b>  |
| rl5915       | <b>1</b>   | <b>388,258</b>  | 0          | -               | 0          | -                |
| rl5934       | <b>1</b>   | <b>269,927</b>  | 0          | -               | 1          | 2112,233         |
| u1060        | 7          | 548,345         | <b>11</b>  | <b>1354,223</b> | 11         | 1439,33          |
| u1432        | 4          | 545,712         | <b>10</b>  | <b>3070,881</b> | 10         | 3132,115         |
| u159         | 11         | 10,47           | 11         | 67,248          | <b>11</b>  | <b>40,044</b>    |
| u1817        | 4          | 71,366          | 7          | 6962,838        | <b>7</b>   | <b>6101,453</b>  |
| u2152        | 4          | 83,851          | 5          | 2620,111        | <b>5</b>   | <b>2568,356</b>  |
| u2319        | 2          | 40,895          | 9          | 8287,439        | <b>9</b>   | <b>7926,615</b>  |
| <b>TOTAL</b> | <b>123</b> | <b>14784,52</b> | <b>166</b> | <b>47037,91</b> | <b>173</b> | <b>57392,109</b> |

TABLE 1 – Comparaison entre les méthodes indépendantes et l’approche portfolio. La meilleure méthode pour chaque famille d’instances est indiquée en gras.

l’ensemble {5, 10, 20, 30, 40, 50, 100, 200, 300, 400, 500}.

La figure 1 montre la proportion des problèmes de décision résolus en fonction du nombre d’itérations restantes dans l’algorithme 1. Par exemple, si la courbe passe par le point (5, 0.6), cela signifie que 60 % des instances n’avaient besoin que de 5 itérations restantes dans l’algorithme 1 pour atteindre l’optimum. Ainsi, nous observons naturellement une baisse de la proportion à mesure que le nombre d’itérations diminue. On remarque que le solveur MaxCDCL a une meilleure performances pour les premières itérations, c’est-à-dire typiquement lorsque plus de 12 itérations restent pour atteindre l’optimum. En fait, nous avons constaté que, pour les premières itérations de l’algorithme, MaxCDCL a une meilleure performance comparé à CPLEX pour presque 100% des instances résolues. Par conséquent, dans notre méthode portfolio, nous commençons par résoudre chaque problème DS en utilisant la formulation MaxSAT avec MaxCDCL lorsqu’il reste plus de 12 itérations, puis nous terminons les calculs restants avec le programme linéaire résolu par CPLEX. À chaque itération, le problème DS est soit résolu, soit il ne l’est pas après une heure et l’algorithme s’arrête, sinon l’algorithme continue et peut potentiellement atteindre l’optimum pour l’instance traitée du problème de p-centre. Nous limitons le temps de calcul global pour toutes les itérations à 1 heure.

Les résultats obtenus par les différentes méthodes sont rapportés dans le tableau 1. Pour chaque instance de TSPLIB, nous reportons le nombre d’instances ré-

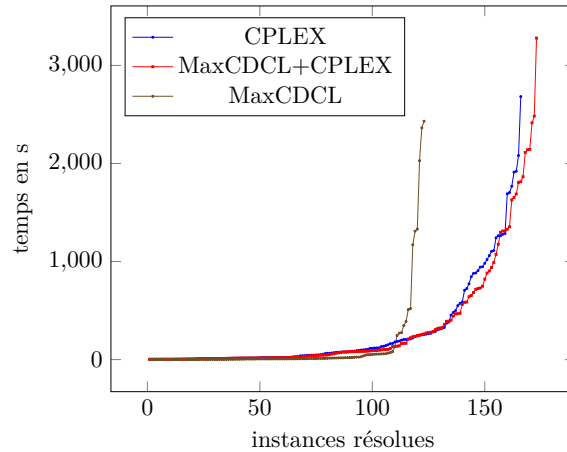


FIGURE 2 – Nombre d’instances résolues en fonction du temps CPU

solues (sur 11) et le temps de résolution cumulatif en secondes. Nous observons clairement que notre méthode portfolio surpasse les approches MaxSAT et LP puisqu’elle résout 7 instances de plus (+3%) par rapport à la meilleure méthode. La combinaison semble également très compétitive en termes de temps de résolution par rapport à LP, comme le montre la figure 2. Enfin, nous mentionnons que nous avons testé un modèle de décision de satisfiabilité (SAT) (résolu par le solveur Glucose), assez similaire à celui décrit à la section 3, et celui-ci a réussi à résoudre 123 instances mais avec un temps cumulatif de 22923,7 secondes, ce qui corrobore notre choix de MaxSAT comme un paradigme de résolution pertinent même pour les problèmes de décision. Ces observations s’alignent avec une tendance récente consistant à appliquer des paradigmes d’optimisation pour résoudre des problèmes de décision dans le contexte de la satisfiabilité, comme par exemple dans [1].

## 5 Conclusion

Cet article présente une approche novatrice pour résoudre le problème des p-centres en exploitant une formulation MaxSAT. La méthode portfolio proposée, intégrant à la fois une formulation de programme linéaire existante et la nouvelle formulation MaxSAT, démontre des améliorations de performance notables. Plus précisément, notre algorithme montre une augmentation de 3% des instances résolues par rapport aux méthodes existantes. Pour les travaux futurs, nos efforts seront dirigés vers l’amélioration de la performance de notre méthode portfolio. Cela implique d’incorporer des modèles et solveurs plus diversifiés et de se pencher sur le raffinement des paramètres de l’algorithme afin d’obtenir des résultats supérieurs.

## Références

- [1] C. ANSÓTEGUI et J. LEVY : Reducing SAT to max2sat. In Z. ZHOU, éd. : *IJCAI 2021*, p. 1367–1373. ijcai.org, 2021.
- [2] H. CALIK, M. LABBÉ et H. YAMAN : p-Center Problems. In G. LAPORTE, S. NICKEL et F. Saldanha da GAMA, édés : *Location Science*, p. 79–92. Springer International Publishing, Cham, 2015.
- [3] H. CALIK et B. C. TANSEL : Double bound method for solving the p-center location problem. *Computers & Operations Research*, 40(12):2991–2999, déc. 2013.
- [4] M. S. DASKIN : *Network and discrete location : models, algorithms, and applications*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, New York Chichester, 1995.
- [5] S. ELLOUMI, M. LABBÉ et Y. POCHET : A New Formulation and Resolution Method for the p-Center Problem. *INFORMS Journal on Computing*, 16(1):84–94, fév. 2004.
- [6] R. S. GARFINKEL, A. W. NEEBE et M. R. RAO : The  $m$ -Center Problem : Minimax Facility Location. *Management Science*, 23(10):1133–1142, juin 1977.
- [7] S. L. HAKIMI : Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. *Operations Research*, 12(3):450–459, juin 1964.
- [8] P. ILHAN : An efficient exact algorithm for the vertex p-center problem. *Bilkent University Technical Report*, sept. 2001.
- [9] O. KARIV et S. L. HAKIMI : An algorithmic approach to network location problems. i : The p-centers. *SIAM journal on applied mathematics*, 37(3):513–538, 1979.
- [10] C. LI, Z. XU, J. COLL, F. MANYÀ, D. HABET et K. HE : Combining clause learning and branch and bound for maxsat. In L. D. MICHEL, éd. : *CP 2021*, vol. 210 de *LIPICs*, p. 38 :1–38 :18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- [11] 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.
- [12] 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.
- [13] X. LIU, Y. FANG, J. CHEN, Z. SU, C. LI et Z. LU : Effective Approaches to Solve  $P$ -Center Problem via Set Covering and SAT. *IEEE Access*, 8:161232–161244, 2020.
- [14] E. MINIEKA : The  $m$ -center problem. *Siam Review*, 12(1):138–139, 1970.