New robust and probabilistic models for shelter location in forest fire context

Marcel Adonis Haddad

▶ **To cite this version:**

Marcel Adonis Haddad. New robust and probabilistic models for shelter location in forest fire context. Operations Research [math.OC]. Université Paris sciences et lettres; RMIT University (Melbourne), 2020. English. ⟨NNT : 2020UPSLD021⟩. ⟨tel-03259596⟩

HAL Id: tel-03259596

https://theses.hal.science/tel-03259596v1

Submitted on 14 Jun 2021

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.

HAL Authorization

PSL UNIVERSITÉ PARIS

RMIT UNIVERSITY

THÈSE DE DOCTORAT

DE L'UNIVERSITÉ PSL

Préparée à Université Paris Dauphine - PSL
Dans le cadre d'une cotutelle avec RMIT University

Nouveaux modèles robustes et probabilistes pour la localisation d'abris dans un contexte de feux de forêt

New robust and probabilistic models for shelter location in forest fire context

Soutenue par
Marcel Adonis Haddad
Le **14/12/2020**

École doctorale n°543
SDOSE

Spécialité
Informatique

Composition du jury :

DELLA CROCE Federico Politecnico Di Torino	*Rapporteur*
DEMANGE Marc Royal Melbourne Institute of Technology	*Directeur de thèse*
ELLOUMI Sourour ENSTA Paristech	*Rapporteur*
MURAT Cécile Université Paris Dauphine - PSL	*Directrice de thèse*
PASCHOS Evangelios Université Paris Dauphine - PSL	*Président du jury*
QUADRI Dominique Université Paris-Sud	*Examinateur*
WIDMER Marino Université de Fribourg	*Examinateur*

Dauphine | PSL

RMIT UNIVERSITY

New robust and probabilistic models for shelter location in wild fire context

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

Marcel Adonis Haddad

Master's degree in Computer Science Engineering, Université de Technologie de Compiègne
Master's degree in Mathematics, Computer Science, Decision and Organisation, Université Paris Dauphine - PSL, Mines ParisTech

School of Science

College of Science, Technology, Engineering and Maths

RMIT University

October, 2020

CANDIDATE DECLARATION

The work contained in this joint PhD thesis undertaken between RMIT and Universite Paris-Dauphine has not been previously submitted to meet requirements for an award at these or any other higher education institutions. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made. I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

(Marcel Adonis Haddad — 29 October 2020)

REMERCIEMENTS

Je tiens à remercier très chaleureusement mes encadrants, Cécile Murat et Marc Demange, et à leur exprimer toute ma gratitude. Vous m'avez assuré un environnement de travail agréable, autonome et sécurisant. Merci pour vos conseils, votre professionnalité et votre amitié. Merci Cécile pour ta disponibilité à tout moment. Merci Marc pour l'accueil et le temps passé ensemble à Melbourne. Je repenserai surement avec nostalgie à nos réunions skype/teams/framatalk matins et soir, à adapter selon les différences de fuseaux horaires du moment. Merci à Virgine Gabrel également, pour les séances de travail passées avec Cécile et les fous rire, entre autres sous le soleil andalou.

Une thèse c'est un encadrement, et c'est aussi un environnement propice pour l'esprit et le corps. Je tiens à remercier mes proches, en premier lieu mes parents, qui se faisaient du soucis pour mon bac. Je vous dédie cette thèse. Je tiens à remercier particulièrement ma mère, qui m'a soutenu en cette dernière année de retraite et de confinement covid. Merci à vous deux, Petra et Alexandre, pour l'amitié et l'adelphité qui nous lie et qui me rassure. Merci à mes ami.e.s, ma famille choisie, qui avez partagé le plus mon quotidien et mes questionnements d'une certaine manière. Merci Floriane, Titi, Nasr, Rémi, Noémie, Diane, Marine, Anais, Luce, Thibaut, Julie, Louise, Cornelia, Karim, Celine, Elma, Fantin, Noémie, Pleu et les autres, de toute façon vous n'allez pas lire ces lignes. Merci à tous.tes celleux avec qui j'ai passé des bons moments tout simplement.

Merci aux camarades du Lamsade. Merci à Vangelis Paschos de m'avoir invité à boire un café il y a 5 ans de cela. Je ne sais pas si je serai venu faire un master de recherche à Dauphine sans cette invitation, et du coup l'opportunité d'effectuer cette thèse. Merci aux collègues et amis doctorants, pour nos débats et l'ambiance studieuse au laboratoire. Justin, Ian, George, Mehdi et Mehdi, Hiba, Thomas, Axel, Diana, Amin, Khalil, Beatrice, Ioannis, Celine, Aude, Maude,

Olivier, Annaelle et j'en oublie surement, ça fait trop longtemps que l'on ne s'est plus vus. Merci à tous mes collègues du Lamsade, je suis content d'avoir été membre de ce laboratoire à vos côtés. Merci aux personnes qui m'ont conseillé il y a des années de cela, quand j'étais fraîchement diplomé de l'UTC. Je pense en particulier à Taha Araboui et Veronique Misseri.

Enfin, j'aimerai remercier le compositeur Joe Hisaishi, dont la musique m'a tant accompagné durant mon travail. Elle m'a apporté le sourire et la volupté dans les moments les plus difficile.

SUMMARY

Due to the effects of global warming, the increase in number and in severity of forest fires, or wild fires, is a matter of concern around the world. Natural disasters like forest fires require an emergency management. Emergency management consists in a continuous process with four key components: mitigation & prevention; preparedness; response; and recovery. In this context, we address the problem of locating shelters to mitigate the casualties caused by a fire. We propose new shelter location models as two-phase problems, where the first phase incorporates elements of prevention and preparedness, while the second phase corresponds to the response.

A shelter is a safe place that provides cover from the fire. They can reduce the evacuation time and thus the number of casualties, and they allow to save resources that would otherwise have to be employed to ensure people safety. Recent studies have highlighted the need to develop new models for the location of fire shelters which was confirmed in discussions with final users as part of the GEO-SAFE project.

The problems of determining the "best" location for a set of facilities in order to serve a set of demand points are generally referred as *Facility Location* problems. These problems lie at the core of the *Location Science* research area in the field of Operations Research, and more precisely of Combinatorial Optimization. The meaning of "best" depends on the nature of the problem under study, in particular the constraints and of the optimality criteria considered. Problems where the input data available to decision-makers are incomplete, or non deterministic at the time of decision-making, are said under *indeterminacy*.

We want to locate a limited number of shelters on a territory where the general objective is to minimize the risk of having people trapped by the flames, i.e., to maximize their chance to reach a safe shelter before the fire gets too close. We

represent the territory by a graph. The uncertainty of fire outbreaks is introduced taking into account a finite set of fire scenarios. A scenario defines a fire outbreak on a single zone with the main consequence of modifying the operational paths.

At most p shelters are to be located on vertices in the preparedness phase, thus our solution is a p-set. In the response phase, that is when a scenario occurs, everyone in the territory must reach a shelter following an evacuation path. The main novelty of our model is the Under pressure evacuation strategy. With this evacuation strategy, we propose a robust representation of the evacuation of the source vertex: as there may be different people in different places within the source vertex, and given the pressure induced by the imminent danger and the disruptions caused by the fire on the internal transport network in the source vertex, we consider that any of the paths leaving the vertex may be used for evacuation.

We introduce two problems with the Under Pressure evacuation strategy, which differ by the indeterminacy environment considered. We introduce then two new facility location problems, incorporating the UP evacuation strategy, as non deterministic variants of the classic`Min p-Center` problem. The `PpCP` problem for a risk environment, and the `RpCP` problem for an uncertainty environment.

In our thesis, we analyze the hardness of `RpCP` and `PpCP`. We present NP-hardness results for `RpCP` and non-approximation results for `PpCP`. We take care of establishing hardness results on classes of graphs with practical significance, like planar graphs or subgrids. Then we present polynomial approaches to generate exact or approximation results for `PpCP` and `RpCP`. We use auxiliary subproblems that reveal to be interesting on their own. Finally, we present different mathematical methods to compute exact solutions of `RpCP` and `PpCP`. In particular, we propose an efficient algorithm for `RpCP`, tractable on large instances as shown by our experimental results.

RÉSUMÉ

A cause du réchauffement climatique, le nombre et l'intensité des feux de forêts augmentent autour du globe. Ces feux provoquent d'importants dommages sociaux-économiques et écologiques. De nombreuses études récentes soulignent la nécessité d'élaborer de nouvelles politiques qui tiennent compte des interrelations complexes et dynamiques entre les facteurs sociaux, économiques, environnementaux et politiques. Des campagnes de sensibilisation doivent être développées sur le risque d'incendie dans toutes les régions touchées afin que les individus et les collectivités puissent se préparer à ce type d'événements. De plus, la communauté scientifique développe de nouvelles méthodes et de nouveaux outils pour aider à limiter les impacts des incendies sur nos sociétés et notre environnement. Cette thèse, organisée en cotutelle entre Paris-Dauphine Université et RMIT University, s'inscrit dans le cadre du projet "Geospatial based Environment for Optimization Systems Addressing Fire Emergencies", GEO-SAFE financé par Horizon 2020 Marie Sklodowska-Curie RISE. Le projet GEO-SAFE vise à proposer des méthodes innovantes dans la gestion de problèmes liés aux incendies de forêt. Les situations d'urgence liées aux catastrophes naturelles se caractérisent par un environnement décisionnel difficile, des autorités publiques confrontées à des délais serrés, des ressources limitées, une énorme quantité de données à analyser, une incertitude extrême et des objectifs multiples qui peuvent entrer en conflit ainsi que de fortes contraintes lorsqu'il s'agit de traiter des questions de sécurité et de sauver des vies. La gestion des urgences consiste en un processus continu comportant quatre éléments clés : la prévention et l'atténuation, la préparation, l'intervention et le rétablissement. Dans ce contexte, nous abordons le problème de la localisation d'abris anti-feu, ou refuges, dans le but d'atténuer le plus possible le nombre de victimes d'un incendie. Un refuge est un bâtiment équipé pour protéger les personnes contre les flammes, mais il peut aussi s'agir d'un point de rassemblement d'urgence

sécurisé. Les autorités publiques ont manifesté un intérêt croissant pour ce type de refuge. Bien que l'idée ne soit pas nouvelle, son efficacité dépend fondamentalement de la méthode de localisation des refuges : comme le territoire à couvrir par un ou plusieurs refuges peut être arbitrairement grand, le décideur doit prendre en considération les voies et le temps nécessaire aux personnes dispersées sur le territoire pour atteindre un refuge. Il est en effet nécessaire de garantir des conditions d'évacuation sûres pour les randonneurs et autres utilisateurs de la forêt. Comme les incendies et leurs conséquences sont par nature imprévisibles, il peut être difficile de localiser les refuges de façon adaptée à chaque configuration possible de feux. Des études récentes ont mis en évidence la nécessité de développer de nouveaux modèles pour la localisation des refuges ce qui a été confirmé lors des discussions avec les utilisateurs finaux dans le cadre du projet GEO-SAFE. Nous proposons de nouveaux modèles non-déterministes de localisation des refuges en deux phases. La première phase comporte des éléments de prévention et de préparation. La deuxième phase décrit l'intervention déclenchée par un départ de feu. Ces problèmes s'inscrivent dans le domaine plus large des problèmes de localisation (en anglais *Facility Location*), domaine appartenant au champ de la recherche opérationnelle et de l'optimisation combinatoire. L'approche non-déterministe permet de mieux appréhender l'incertitude et le manque de données inhérents aux problèmes liés aux feux de forêts. Dans ce cadre, notre objectif est de localiser sur un territoire un nombre limité de refuges, de sorte à minimiser le risque que des individus soient piégés par des flammes.

Nous considérons dans cette thèse un territoire avec une faible densité d'individus présents (résidents ou utilisateurs de la forêt). Le territoire est représenté par un graphe connexe non-orienté. Chaque sommet du graphe correspond à une zone, et deux sommets sont connectés par une arête pondérée si on peut aller d'une zone à l'autre. Une longueur peut-être associée à chaque arête, celle-ci corresponds alors à la distance d'une zone à l'autre ou à la durée nécessaire pour se déplacer d'une zone à l'autre. Quand un feu se déclare sur une zone, nous appelons le sommet correspondant le sommet source. Nous utilisons des scénarios pour décrire les possibles évolutions de l'état du système causés par un départ de feu. Nous considérons que lorsqu'un feu de forêt se déclare, il existe des systèmes d'alerte rapides et efficaces (mais non personnalisés, par exemple des sirènes) pour appeler tous les individus présents sur le territoire à évacuer vers un refuge. Dans cette thèse nous ne prenons pas en compte la capacité des refuges vu que, dans notre contexte, les refuges sont bâtis avec des capacité relativement très larges. Dans cette thèse, nous considérons des scénarios où le feu est restreint à une zone. En effet, une fois l'alarme donnée, nous considérons que les individus doivent évacuer dans un délai de temps très court, avant que le feu ne se propage au-delà de sa zone de départ. Ces hypothèses dépendent de la méthode de modélisation du ter-

ritoire en zones. Notre problème peut alors être décrit comme un cas particulier de problèmes de localisation-allocations. Une solution décrit une localisation de refuge sur les graphe, qui correspond à un sous-ensemble de sommets. De plus, une solution induit, pour chaque scénario et pour les individus présents sur une zone, une possible allocation à refuge à rejoindre. Plus précisément, une allocation correspond à un chemin d'évacuation. On appelle stratégie d'évacuation les actions attendues prises par les individus évacuant le territoire lorsqu'un feu se déclare. La stratégie d'évacuation induit les chemins d'évacuations qui seront empruntés par les individus selon leur position et le scénario. La principale contribution de notre thèse est de proposer la nouvelle stratégie d'évacuation nommée *Under Pressure*, qui décrit plus fidèlement les décisions prises sous pression du danger. La principale originalité de cette stratégie est la suivante : si aucun refuge n'est localisé sur le sommet source, alors on considère qu'il existe un chemin d'évacuation vers chaque sommet adjacent. Cette hypothèse de modélisation peut-être doublement justifiée : le feu dans la zone source peut restreindre la mobilité interne, interdisant d'accès certaines voies de sorties ; la proximité, l'intensité et l'immédiateté du danger peut affecter la prise de décision des individus sur cette zone. Dans les 2 cas, on ne peut déterminer la direction d'évacuation immédiate choisie par les individus sur la zone source, nous devons donc considérer que n'importe quelle arête sortante peut être empruntée. Pour tous les autres sommets du graphe, un chemin d'évacuation est un plus court chemin entre ce sommet et le plus proche refuge. On appelle distance d'évacuation du sommet la longueur maximum d'un de ses chemins d'évacuation. Notre stratégie d'évacuation induit des distances d'évacuations particulières qui marquent la spécificité de notre modèle. Nous introduisons deux problèmes s'appuyant sur la stratégie d'évacuation Under Pressure, qui diffèrent par l'environnement non-déterministe considéré. Nous présentons ces deux problèmes comme des variantes non-déterministe du problème déterministe classique du `Min p-Center`: le `Min p-Center` est le problème de trouver un ensemble de au plus p sommets, appelés *centres*, tel que la distance d'un sommet du graphe à un plus proche centre soit minimale. Dans notre cas les centres correspondent aux refuges.

Sous la stratégie d'évacuation Under Pressure, la valeur d'une solution pour un scénario donné est le rayon d'évacuation, dont la valeur est donnée par la distance d'évacuation maximum d'un sommet pour ce scénario. Nous définissons deux nouveaux problèmes sur la base de deux catégories d'environnements non-déterministe : l'environnement d'incertitude et l'environnement de risque. Dans un environnement de risque, la valeur de certains paramètres non-déterministes suivent des lois de probabilité connues par le décideur. Cette configuration s'inscrit alors dans le cadre de l'optimisation stochastique ou probabiliste. Le *Probabilist p-Center Under Pressure* noté `PpCP` est le problème de localiser au plus p refuges

de sorte à minimiser le rayon d'évacuation moyen sur tous les scénarios. Dans un environnement d'incertitude, aucune information probabiliste n'est connue ou utilisée pour les paramètres non-déterministes. Cette configuration s'inscrit alors dans le cadre de l'optimisation robuste. Le *Robust p-Center Under Pressure* noté `RpCP` est le problème de localiser au plus p refuges de sorte à minimiser le rayon d'évacuation maximum sur tous les scénarios. Voici une liste des fortes hypothèses de notre modèle: Dans le chapitre 2, nous décrivons en détail notre modèle. Nous présentons nos hypothèses de modélisation pour représenter le territoire, les refuges et l'incertitude liée aux départs de feu. Nous introduisons les problèmes du `RpCP` et du `PpCP` et l'ensemble des solutions réalisables pour ces 2 problèmes. Nous comparons également nos problèmes à certains travaux pertinents, relevant de la littérature sur le sujet. Dans le chapitre 3, nous analysons la NP-difficulté des problèmes `RpCP` et `PpCP`. Nous proposons différentes réductions de problèmes déterministes classiques. Nous présentons également des résultats de non-approximation pour le `PpCP`. Nous proposons des résultats sur des classes de graphes pertinentes pour notre problème, comme la classe des graphes planaires ou les sous grilles. Dans le chapitre 4, nous présentons des approches polynomiales pour générer des solutions exactes ou approximables pour le `PpCP`. Nous proposons une solution explicite pour les chemins avec des longueurs d'arêtes uniforme en utilisant des problèmes auxiliaires. La preuve est étonnamment non triviale. Nous présentons également un algorithme pour `PpCP` garantissant un ratio d'approximation fixe sur les graphes de degrés bornés et de longueur d'arêtes bornée. Dans le chapitre 5, nous proposons des algorithmes exacte à temps polynomial pour `RpCP` sur différentes classes de graphes acycliques aux arêtes pondérées. Nous utilisons des propriétés structurelles identifiées dans le chapitre précédent pour introduire une méthode basée sur une recherche dichotomique. Nous proposons des algorithmes pour les chemins, les étoiles, les chenilles, et les arbres. Dans le chapitre 6, nous proposons des méthodes mathématiques basées sur la programmation linéaire pour résoudre le `RpCP` et du `PpCP`. Ces méthodes sont des généralisations étendues de programmes linéaires existants pour le problème du `Min p-Center`, qui permettent de prendre en compte l'indétermination et la stratégie l'évacuation Under Pressure.

Présentation des Problèmes

Dans ce chapitre, nous définissons les problèmes étudiés dans cette thèse. Notre objectif est de déterminer le meilleur emplacement pour les refuges dans un territoire menacé par les incendies.

Modélisation

Nous représentons le territoire par un graphe connexe $G = (V, E)$ avec $V = \{1, \ldots, n\}$ l'ensemble de sommets, et E l'ensemble des arêtes pondérées. Pour $x, y \in V$, nous dénotons $d(x, y)$ la distance entre les sommets x et y dans G i.e., la longueur d'un plus court chemin entre x et y. Selon notre modèle, les refuges doivent être placés sur des sommets. Les emplacements des refuges correspondent alors à un sous-ensemble $C \subseteq V$. Nous utilisons des scénarios pour modéliser les risques de catastrophes. Un ensemble de scénarios S est une description des états du système après qu'un feu se soit déclaré. Nous considérons l'ensemble des scénarios correspondant à un seul sommet en feu, et nous désignons le scénario associé à un départ de feu sur le sommet $s \in V$ comme le scénario $s \in S$. Le *graphe opérationnel* associé au scénario s, dénoté $G^s = (V^s, E^s)$, est obtenu à partir de G comme suit: toute les arêtes dans E sont gardées dans E^s à l'exception des arêtes $(s, v), v \in V$ adjacentes au sommet s. Ces arêtes sont remplacées par des arêtes orientées (s, v) (appelés aussi arcs) sortantes de s. Pour $x, y \in V$, nous dénotons $d^s(x, y)$ la distance entre les sommets x et y dans G^s. Si aucun chemin existe entre x et y dans G^s, on note $d^s(x, y) = +\infty$.

Une solution décrit également, pour chaque scénario et pour les personnes d'une zone donnée, un ou plusieurs refuge(s) à atteindre sur la base d'une stratégie d'évacuation. La stratégie d'évacuation décrit les mesures que prennent les personnes qui évacuent un territoire une fois qu'un feu se déclare. Cette stratégie d'évacuation doit essayer de prendre en compte les actions que les personnes sur place sont le plus susceptibles d'entreprendre. Nous proposons la stratégie *Under Pressure*, noté UP. Dans la stratégie UP, nous considérons que, lorsqu'un feu se déclare, tout les individus du territoire doivent rejoindre un refuges. Dans le scénario s, nous avons alors trois cas:

- Pour les individus sur le sommet s, nous avons 2 possibilités:
  - Si un refuge est localisé sur le sommet s, nous considérons que les individus peuvent s'y réfugier. Cependant, ce refuge est inaccessible pour tout individus en dehors de cette zone.
  - Sinon, nous considérons que les individus fuiront dans un premier temps dans n'importe quelle direction pour atteindre une zone adjacente j. Une fois arrivés à la zone j, ils se dirigeront alors vers un refuge accessible parmi les plus proches depuis j dans G^s.
- Pour les individus sur les autres sommets, nous considérons que les individus se dirigent vers un refuge accessible parmi les plus proches dans G^s.

Pour garantir un niveau de risque acceptable, nous envisageons le scénario le plus défavorable. La *distance d'évacuation* d'un sommet v dans le scénario s est alors

la longueur maximum d'un chemin d'évacuation de v dans le scénario s. Cette valeur est donnée par:

$$r^s(C,v) = \begin{cases} 0 & \text{si } v = s \text{ et } v \in C \\ \max\limits_{j \in \Gamma(v)} \{l_{v,j} + d^s(j,C)\} & \text{si } v = s \text{ et } v \notin C \\ d^s(v,C) & \text{si } v \neq s \end{cases}$$

Fonctions Objectives

Pour un scénario s, la qualité d'une solution est déterminée sur la base de la distance d'évacuation maximume induite par cette solution pour ce scénario. Nous définissons le *rayon d'évacuation* d'un ensemble $C \subset V$ pour le scénario s par:

$$r^s(C) = \max_{x \in V} r^s(C,x)$$

Nous proposons deux fonctions objectives, s'inscrivant dans deux environnements non-déterministes distincts.

Dans une approche robuste, aucune information sur la probabilité des scénarios n'est utilisée. Nous proposons dans ce cadre la fonction objective $\mathbf{R}$ de type Minimax, donnée par:

$$\mathbf{R}(C) = \max_{s \in V} r^s(C)$$

$\mathbf{R}(C)$ est nommé le *rayon robuste* de C.

Robust p-Center Under Pressure (`RpCP`)

Instance:	Un graphe aux arêtes pondérées $G = (V,E)$, un ensemble de scenarios S, un entier p
Feasible solutions:	Un sous-ensemble $C \subseteq V$ de taille au plus p, avec un rayon robuste fini.
Objective:	Trouver une solution de rayon robuste minimum, si elle existe.

`RpCP` est étroitement lié à un autre problème d'optimisation qui nous sera utile par la suite.

`k-RCP`

Instance:	Un graphe aux arêtes pondérées $G = (V,E)$, un ensemble de scenarios S, un entier k
Feasible solutions:	Un sous-ensemble $C \subseteq V$, avec $\mathbf{R}(C) \leq k$.
Objective:	Trouver une solution réalisable de taille minimum.

Dans une approche probabiliste, une probabilité π_s est assignée à chaque scénario s. Nous proposons dans ce cas la fonction objective $\mathbf{E}$, donnée par:

$$\mathbf{E}(C) = \sum_{s \in V} \pi_s \times r^s(C)$$

$\mathbf{E}(C)$ est dit le *rayon probabiliste* de C. Dans cette thèse, nous nous intéressons uniquement à des scénarios avec une probabilité uniforme i.e., $\forall s \in V, \pi_s = \frac{1}{n}$.

Probabilistic p-Center under Pressure (`PpCP`)

Instance:	Un graphe aux arêtes pondérées $G = (V, E)$, un ensemble de scenarios S, un entier p
Feasible solutions:	Un sous-ensemble $C \subseteq V$ de taille au plus p, avec un rayon probabiliste fini.
Objective:	Trouver une solution de rayon probabiliste minimum, si elle existe.

Par la suite, nous dénotons (G, p) une instance du problème `RpCP` ou `PpCP`, l'ensemble des scénarios (et leur probabilité dans le cas probabiliste) étant défini en fonction du graphe.

Conditions de faisabilité

Une solution C est réalisable pour `PpCP` et `RpCP` si $r^s(C)$ est fini pour tout scénario s. Une instance (G, p) du problème `PpCP` ou `RpCP` n'admet pas systématiquement de solutions réalisables. Nous présentons dans cette partie les conditions nécessaires et suffisantes pour qu'une solution sur une instance (G, p) du problème `PpCP` ou `RpCP` soit réalisable.

Soit G un graphe connexe composé d'au moins 2 sommets. Une composante d'articulation de G est une composante connexe du graphe induit par la suppression d'un point d'articulation de G. Une composante d'articulation minimale, dénotée MAC, est une composante d'articulation ne contenant aucune autre compostante d'articulation. Nous montrons qu'une solution n'est réalisable pour `RpCP` et `PpCP` si et uniquement si $2 \leq |C| \leq p$ et C contient au moins un sommet de chaque composante d'articulation minimale de G.

Il s'ensuit que, pour tout graphe G, $(G, 1)$ n'admet pas de solution. Pour $p \geq 2$, nous pouvons vérifer en temps polynomial si (G, p) admet des solutions. Enfin, un dernier corollaire est la nécessité pour toute solution réalisable de (G, p) d'inclure tout sommet pendant (sommet de degré 1).

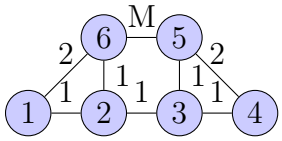


Figure 1: Example d'une instance avec une solution optimale différente pour `Min p-Center` et `RpCP` pour $p = 2$.

Comparaison avec `Min p-Center`

Considérons le graphe de la figure 1. Pour $p = 2$, on observe que la solution optimale pour `Min p-Center` est la solution $C = \{2, 3\}$ de rayon 1. Le rayon robust de C est de $M + 3$, induit par les scénarios 2 et 3. Le rayon probabiliste de C est également de l'ordre de M. Dans le même temps, la solution $D = \{5, 6\}$, de rayon 2, assure néanmoins un rayon robuste de 3 et un rayon probabiliste de 16/6. Globalement on observe donc que la solution optimal pour `Min p-Center` peut engendrer une erreur relative d'ordre M.

Enfin, dans la deuxième partie du chapitre 2, nous comparons les problèmes `RpCP` et `PpCP` à des problèmes pertinent de la littérature existante s'inscrivant dans le même cadre non-déterministe. À notre connaissance, il s'agit de la première tentative de modélisation du processus d'évacuation de cette manière pour un problème de localisation de refuges. Ce modèle se différencie de la plupart des modèles existants en ce que nous appliquons l'indétermination sur la structure du graphe et non sur la longueur des arêtes. Ce faisant, nous considérons que nous pouvons mieux appréhender l'impact de l'incendie sur les distances d'évacuation.

Résultats de NP-difficulté

Dans ce chapitre, nous analysons la NP-difficulté des problèmes `RpCP` et `PpCP`. Nous considérons la version du `RpCP` avec un seuil de rayon d'évacuation constant, $\texttt{RpCP}_k$. Nous donnons différentes réductions de `Min Vertex Cover` et `Min Dominating Set` à $\texttt{RpCP}_k$ pour différentes valeurs de k. Il s'ensuit que :

- sur toutes les classes héréditaires de graphes pour lesquelles la version décisionnelle du problème `Min Vertex Cover` est NP-complète : $\texttt{RpCP}_1$ est NP-complet et `RpCP` est NP-difficile.
- sur toutes les classes héréditaires de graphes pour lesquelles la version décisionnelle du problème `Min Vertex Cover` peut être résolu en temps polynomial : $\texttt{RpCP}_1$ est résoluble en temps polynomial.

- sur toutes les classes de graphes uniformes sans triangle et de degré minimum 2, pour lesquelles la version décisionnelle du problème `Min Dominating Set` est NP-complète : $\texttt{RpCP}_2$ est NP-complet et `RpCP` est NP-dur.

- dans toutes les classes de graphes uniformes sans triangle et de degré minimum 2, pour lesquelles la version décisionnelle du problème `Min Dominating Set` est résoluble en temps polynomial : $\texttt{RpCP}_2$ est résoluble en temps polynomial. $\texttt{RpCP}_2$ est résoluble en temps polynomial. $\texttt{RpCP}_2$ est résoluble en temps polynomial. $\texttt{RpCP}_2$ est résoluble en temps polynomial. $\texttt{RpCP}_2$ est résoluble en temps polynomial.

Pour le cas robuste, notre résultat le plus solide, soit que `RpCP` est NP-difficile dans les sous-grilles avec des sommets de degré 2 ou 3, est obtenu par une réduction de `Min Vertex Cover` à $\texttt{RpCP}_2$, par le biais de la relation entre $\texttt{RpCP}_2$ et `Min Dominating Set`. En outre, nous avons souligné l'existence de classes de graphes sur lesquelles $\texttt{RpCP}_1$ est difficile mais $\texttt{RpCP}_2$ est trivial. Cela semble confirmer qu'une réduction de $\texttt{RpCP}_k$ à $\texttt{RpCP}_k + 1$ n'existe pas dans le cas général et, par conséquent, la difficulté de $\texttt{RpCP}_k$ sur une classe de graphe donnée doit être étudiée pour toute valeur de k et ne peut être déduite, en général, des résultats de difficulté avec différentes valeurs de k.

En ce qui concerne `PpCP`, nous donnons deux résultats de non approximation. Premièrement, nous prouvons qu'il n'y a pas d'approximation polynomiale du temps pour `PpCP` garantissant un rapport inférieur à $\frac{20}{19}$ pour les graphes planaires bipartites de degrés 2 ou 3. En particulier, dans la section 3, nous réexaminons la réduction précédente pour prouver que `PpCP` n'est pas approximable avec un rapport inférieur à $\frac{56}{55}$ sur des sous-grilles de degré au plus 3. Même si ce dernier résultat ne généralise pas celui que nous avons obtenu précédemment (la classe est plus restrictive mais la nouvelle limite est plus proche de 1), la preuve nécessite une analyse beaucoup plus approfondie.

Dans le chapitre suivant, nous proposons des méthodes polynomiales pour générer des solutions exactes ou garantissant un ratio d'approximation pour `PpCP`.

Résults polynimiaux et d'approximation pour le `PpCP`

Nous avons vu dans le chapitre précédent que `PpCP` est NP-difficile sur les graphes planaires et les sous-grilles de degrés $\{2, 3\}$. Dans ce chapitre, nous proposons des méthodes polynomiales pour calculer des solutions pour `PpCP` avec des garanties de performance. Nous montrons dans la section 4.1 que, sur un chemin dans le cas uniforme, une solution monotone équilibrée est une solution optimale pour `PpCP` (Théorème 30). Même si le résultat n'est pas surprenant, la preuve est

étonnamment non triviale. Ce résultat repose sur une décomposition du problème en sous-problèmes. L'idée principale est d'exprimer la fonction objective comme la somme de deux parties (contribution des sommets avec refuges d'une part et des sommets sans refuges d'autre part) et de simultanément borner la fonction obective par une autre fonction objective plus facilement calculable. Nous introduisons le rayon d'évacuation local d'une solution pour un scénario donné comme étant la distance d'évacuation maximum du sommet source et de ses sommets adjacents. Le rayon d'évacuation local est une limite inférieure au rayon d'évacuation d'une solution. Lorsque la solution est équilibrée, le rayon d'évacuation équivaut au rayon d'évacuation locale. En outre, nous considérons une partition des scénarios en deux ensembles. Pour chaque ensemble de scénarios, nous prouvons indépendamment qu'une solution équilibrée monotone minimise le rayon d'évacuation local. Ce résultat n'est toutefois pas valable sur les graphes aux arêtes pondérées.

Dans la section 4.2, nous proposons des résultats d'approximation pour `PpCP`. Dans les graphes avec des arêtes de degré borné et avec des longueurs d'arêtes inclues dans $[\ell, 2\ell]$, `PpCP` est approximable avec un rapport de $4\overline{deg}(G) + 2$ (où $\overline{deg}(G)$ est le degré moyen du graphe G). Ce résultat est valable pour une classe d'instance sur lesquels tous nos résultats de NP-difficulté s'appliquent. Sur les arbres, `PpCP` est approximable avec un rapport constant de 3.

En outre, nous introduisons le problème `Min MAC p-Center`, qui consiste à trouver un ensemble p de rayon minimum parmi l'ensemble des solutions possibles pour `PpCP` i.e., `Min MAC p-Center` est une variante de `Min p-Center` où l'ensemble de solutions réalisables est restreint à l'ensemble de solutions réalisables pour `PpCP`. Nous montrons que ce problème est approximable avec un ratio de 2 et que c'est le meilleur rapport d'approximation constant possible. Il est également polynomial sur les arbres.

Dans le chapitre suivant, nous considérons le problème `RpCP` où nous généralisons certaines notions introduites dans ce chapitre. En effet, l'idée d'une segmentation induite par une solution sera utilisée et généralisée pour développer des algorithmes polynomiaux pour `RpCP` sur des graphes acycliques avec arêtes pondérées.

Résults polynimiaux et d'approximation pour le `RpCP`

Dans ce chapitre, nous présentons un algorithme générique pour résoudre `RpCP` en se basant sur un algorithme pour `k-RCP` et sur une recherche dichtomique. Considérons l'instance (G, p) de `RpCP`, avec $p \geq 2$ et G un graphe aux arêtes pondérées. Soit D l'ensemble croissant des valeurs candidates pour le rayon robust de G. Nous observons que, pour tout $k \in D$, si la solution optimale pour `k-RCP` est de taille supérieure à p, alors la solution optimale pour `RpCP` a un rayon robuste supérieure à

k. Sur la base de cette relation, nous pouvons effectuer une recherche dichotomique sur D pour trouver le rayon k^* minimum pour lequel la taille de la solution optimale de `k-RCP` est inférieur ou égale à p. Cette méthode est implémentée dans l'Algorithme 2, dans lequel on utilise un algorithme non défini pour résoudre `k-RCP`. La complexité de l'Algorithme 2 dépends de la complexité de l'agorithme utilisée pour résoudre `k-RCP`. Nous proposons des algorithmes polynomiaux de résolution exacte de `k-RCP` sur des classes de graphes acycliques. Ces méthodes efficaces - de type programmation dynamique - reposent sur des propriétées structurelles de ces classes de graphes. Nous proposons plusieurs algorithmes polynomiaux raffinés et en $O(n)$ pour `k-RCP` sur différentes classes de graphes acycliques: les chemins, les étoiles étendues, les chenilles et les arbres. En particulier, comme les refuges doivent être situés sur des sommets pendants, ils constituent une base de départ pour la solution. En partant de ces sommets pendants, nos algorithmes adoptent une approche ascendante pour localiser progressivement les refuges.

Méthodes mathématiques

Dans ce chapitre nous proposons des méthodes de programmation mathématique pour résoudre le `RpCP` et le `PpCP` dans le cas général. Notre objectif est de fournir un algorithme exact pour les deux problèmes même sur les instances pour lesquels il n'existe pas de méthode polynomiale pour résoudre.

Sur la base des formulations de type Mixed Integer Programming (MIP) pour `Min p-Center`, nous proposons trois formulations MIP pour `RpCP` et `PpCP`. Nous proposons également une formulation pour `k-RCP`, baséee sur une formulation en couverture d'ensemble (Set Covering). La taille des instances générées par nos formulations est cependant en pratique trop importante. Afin de proposer une méthode de résolution efficace pour `RpCP`, nous proposons des algorithmes exacts basés sur la formulation P^{rob} et en utilisant un schéma de décomposition. Nous présentons trois méthodes différentes pour calculer les limites inférieure et supérieure de `RpCP`. Pour réduire la taille d'une instance et le nombre d'itérations, nous proposons une généralisation de la recherche binaire : la recherche σ-quantile. Nous testons différents algorithmes, chacun d'entre eux intégrant une méthode différente pour générer les limites inférieure et supérieure. Une de ces méthodes se basent sur une relaxation de P^{rob}, la formulation RP^{rob}, qui correspond à une sous-variante de notre problème pour lequel on ne s'intéresse qu'à l'évacuation du sommet en feu. Nos résultats expérimentaux montrent que l'algorithme EA3* est le meilleur : son efficacité vient de la qualité de la limite inférieure obtenue avec un algorithme de recherche 4-quantile très efficace pour RP^{rob} et du faible temps de calcul. Nous avons testé notre algorithme et généré les différentes limites sur le répertoire d'instances OR et un ensemble domestique de sous-grilles aux longueurs

d'arêtes uniformes et non-uniforme. Dans l'ensemble, l'algorithme basé sur RP^{rob} reste le plus efficace en termes de temps de calcul. En outre, nos expériences mettent en évidence les performances d'une solution optimale `RpCP` par rapport à une solution optimale `Min p-Center`: nous avons comparé la valeur d'une solution optimale pour `Min p-Center` prise pour `RpCP` avec la valeur optimale. Même si nous excluons les cas non réalisables, il apparaît que l'écart peut être assez important, jusqu'à 92% pour des instances avec plus de 100 sommets.

Conclusion

	`RpCP`	`PpCP`
Chemin uniforme	Exact solution*	Exact solution
Chemin, Etoiles, Chenilles non-uniforme	$O(n^2)$	Inconnu
Tree	$O(n^2)$ même dans le cas non-uniforme	3-approximable dans le cas uniforme Inconnu
Graphes avec longueur d'arêtes et degrés bornés.	$(4\overline{deg}(G)+2)$-approximable*	$(4\overline{deg}(G)+2)$-approximable
Graphe bipartite planaire de degré 2 ou 3	NP-difficile	Non approximable sous un ratio de $\frac{20}{19}$
Sous-grille de degré 2 ou 3	NP-difficile	Non approximable sous un ratio de $\frac{56}{55}$
Cas général	NP-difficile Algorithme efficace de résolution basée sur une formulation MIP	NP-difficile Formulation MIP

Table 1: Récapitulatif des principaux résultats pour le `RpCP` et le `PpCP`.

* Ces résultats sur `RpCP` peuvent être directement déduits des resultats correspondants sur `PpCP`.

Notre travail a encore un grand potentiel de développement, tant d'un point de vue théorique que pratique. Nous concluons notre discussion en énumérant quelques pistes possibles de future recherche.

D'un point de vue théorique, une première question ouverte concerne la complexité de la résolution PpCP sur des graphes acycliques pondérés par les bords. Bien qu'on puisse garantir un ratio d'approximation de 3 pour les arbres avec des longueurs d'arêtes non-uniformes mais bornées, nous disposons de peu d'informations sur la structure d'une solution optimale pour PpCP sur des graphes acycliques, ne serait-ce que sur un chemin avec arêtes pondérées. Une piste de recherche serait d'analyser la qualité de la solution optimale de RpCP, calculable en temps polynomial sur ces graphes par rapport à PpCP. Nous savons qu'une solution RpCP optimale n'est pas toujours optimale pour PpCP, cependant nous avons que de timides résultats sur les relations structurelles entre les solutions optimales des deux problèmes. La qualité d'une solution optimale pour RpCP utilisée pour PpCP, et vice versa, devrait être analysée pour différentes classes de graphes.

Une autre piste de recherche pour identifier des solutions pour PpCP serait de restreindre notre problème au rayon probabiliste local, introduit dans la section 4.1. Le rayon probabiliste local, basé sur le rayon d'évacuation local, nous permet de concentrer notre analyse sur la distance d'évacuation maximale induite par le sommet source et son voisinage. La pertinence de cette approche a également été démontrée dans la section 6.4. En effet, RP^{rob} correspond au problème RpCP limité à la distance d'évacuation du sommet source uniquement. Alors que RP^{rob} est une variante détendue du rayon local robuste (la version robuste du rayon probabiliste local), il fournit souvent une limite inférieure étroite pour RpCP. Le rayon d'évacuation local a également un intérêt pratique, sur lequel nous reviendrons plus tard. Enfin, comme l'écart entre notre algorithme d'approximation et nos résultat de non-approximation est encore important, il est possible d'améliorer nos résultats.

Pour RpCP, nous avons commencé à mettre en œuvre nos algorithmes polynomiaux pour les graphes acycliques. Il serait intéressant d'évaluer leur efficacité à l'aide de résultats expérimentaux. En outre, nous n'avons pas eu le temps de concevoir un algorithme d'approximation spécifique pour RpCP sur des graphes et sous-grilles planaires. Un bon point de départ serait de considérer des graphes avec exactement un sommet pendant, et aucun point d'articulation. Dans cette configuration, le problème serait de décider de l'emplacement des autres refuges. Notez que la NP-difficulté de RpCP n'est pas établie sur de tels graphes. Cette approche s'inspire des algorithmes d'approximation des graphes acycliques. En effet, comme les refuges doivent être situés sur les sommets en attente, ces sommets en attente sont la base de nos solutions.

Une autre piste de recherche significative consisterait à améliorer notre algorithme exact basé sur IP pour RpCP et à développer un algorithme exact efficace pour PpCP. Pour ces deux problèmes, nous devrions être en mesure de mettre en œuvre une décomposition de Benders pour résoudre le problème soulevé par la

taille des problèmes. Compte tenu de la qualité observée d'une solution optimale pour RP^{rob} dans le chapitre 6, RP^{rob} peut être considéré comme le problème maître initial. De plus, il serait intéressant de déterminer et de générer des contraintes spécifiques à la classe du graphe considéré.

Une ligne de recherche intéressante serait de considérer les distances vérifiant les inégalités des triangles, et d'analyser si elles induisent des propriétés structurelles pour nos problèmes.

Enfin, certaines hypothèses initiales (propriété des refuges, définition de scénarios, etc.) peuvent être adaptées pour générer différentes variantes de nos problèmes. La pertinence de ces variantes dépend principalement de leur application pratique. D'un point de vue pratique, notre modèle peut être adapté à différentes configurations de feu. Nous en énumérons quelques-unes. Notez que notre modèle de programmation mathématique pourrait facilement intégrer ces variantes, mais nos résultats combinatoires peuvent changer de manière significative.

1. Nous pouvons prendre en compte différentes contraintes ou conséquences de l'implantation d'un refuge sur une zone.

   - Le territoire peut limiter l'emplacement des refuges à certaines zones spécifiques. Dans ce cas, on peut définir dans un premier temps le sous-ensemble des sommets sur lesquels il est possible de localiser un refuge. Cette variante inclut notre modèle comme cas spécifique, ainsi nos résultats de difficulté restent valables. Cependant, des recherches supplémentaires sont nécessaires pour étendre nos résultats de complexité et d'approximation à cette variante.
   - Construire un refuge dans une zone peut impliquer de sécuriser complètement la zone correspondante, auquel cas un incendie ne peut pas se produire dans une zone sur laquelle se trouve un refuge. Nous utilisons un problème auxiliaire relativement proche de ce problème dans la section 4.1, lorsque nous utilisons la fonction objectif $E_{\bar{C}}$. $E_{\bar{C}}$ est la somme du rayon d'évacuation pour tous les scénarios $s \in \bar{C}$, pondérée par le nombre total de scénarios (et non par $|\bar{C}|$). Des recherches supplémentaires sont nécessaires pour réduire cette variante à notre modèle et étendre nos résultats à celui-ci.
   - Au contraire, le refuge situé sur une zone peut ne pas être suffisant pour sécuriser la zone si le feu commence dans la zone. La formule de calcul de la distance d'évacuation peut être adaptée en conséquence. Des recherches supplémentaires sont nécessaires pour réduire cette variante à notre modèle et étendre nos résultats à celui-ci.

2. Nous pouvons prendre en compte la propagation du feu, de manière statique,

en considérant des scénarios associés à un sous-ensemble de sommets en feu. Dans ce cas, tous les sommets en feu ne sont plus atteignables.

3. Dans certaines circonstances, seule une évacuation partielle est nécessaire. Lorsqu'un incendie se produit sur une zone, seule la zone correspondante doit être évacuée. Une variante proche est le cas où seule la zone source et son voisinage doivent être évacués. Cette dernière correspond à l'approche que nous avons initiée dans la Section 4.1 avec le rayon d'évacuation local. Cela renforce notre intérêt pour cette variante.

CONTENTS

Candidate declaration **3**

Remerciements **6**

Summary **8**

Résumé Étendu **23**

Contents **i**

List of figures **v**

List of Tables **vii**

1 Introduction **3**
1.1 Context and Motivations 3
1.2 Basic Notations 8
1.3 Organization of the thesis 13

2 Presentation of the Problems **15**
2.1 The *Under Pressure* model 16
2.1.1 Modeling the territory 17
2.1.2 Scenarios 18
2.1.3 The Under Pressure evacuation strategy 18
2.1.4 Objective functions 23
2.1.5 Feasibility conditions 25
2.1.6 Examples and comparison with *p-Center* 27

2.2 State of the Art . . . 29
2.2.1 The p-Center Problem . . . 29
2.2.2 Models under indeterminacy . . . 34

3 Hardness results **41**
3.1 RpCP Hardness results . . . 43
3.1.1 $\texttt{RpCP}_1$ and the `Min Vertex Cover` problem . . . 44
3.1.2 $\texttt{RpCP}_2$ and the `Min Dominating Set` problem . . . 46
3.1.3 $\texttt{RpCP}_2$ and the `Min Vertex Cover` problem . . . 47
3.1.4 Does increasing the radius make the decision problem harder? 52
3.2 PpCP Non-Approximation Results . . . 54
3.2.1 `PpCP` Non-Approximable on Planar Graphs . . . 54
3.2.2 `PpCP` Inapproximation subgrids . . . 57

4 Polynomial and Approximation results for PpCP **69**
4.1 An explicit solution for PpCP on paths in uniform case . . . 70
4.1.1 Additional notations and preliminary results . . . 70
4.1.2 Auxiliary sub-problems . . . 73
4.1.3 An optimal solution . . . 76
4.1.4 The case of edge-weighted path . . . 81
4.2 Approximation algorithms . . . 83
4.2.1 A polynomial approximation preserving reduction . . . 84
4.2.2 Constant approximation algorithms . . . 87

5 Polynomial and Approximation results for RpCP **97**
5.1 The general approach . . . 98
5.2 Edge-weighted paths . . . 100
5.2.1 Computing the robust radius of a p-set . . . 100
5.2.2 Bounds for `RpCP` . . . 101
5.2.3 Algorithm . . . 101
5.2.4 Example . . . 103
5.3 Extended star with edges of different lengths . . . 105
5.3.1 Computing the robust radius of a p-set . . . 106
5.3.2 Algorithm . . . 107
5.4 Caterpillar . . . 108
5.4.1 Computing the robust radius of a p-set . . . 109
5.4.2 A basic algorithm for `k-RCP` . . . 110
5.4.3 An improved algorithm for `k-RCP` . . . 112
5.5 Trees . . . 116
5.5.1 Computing the robust radius of a p-set . . . 116
5.5.2 Algorithm for `k-RCP` . . . 117

6 Mathematical Programming Approaches **123**
6.1 Integer linear programming models for `p-Center` . . . 125
6.1.1 Daskin's model . . . 125
6.1.2 A set covering based approach . . . 126
6.1.3 The second model . . . 127
6.1.4 Size of the programs . . . 129
6.2 Integer linear programming formulations for `RpCP` . . . 129
6.2.1 MIP formulation for `RpCP` . . . 129
6.2.2 IP formulation for `RpCP` . . . 130
6.2.3 A set covering based algorithm . . . 133
6.2.4 Additionnal inequalities . . . 133
6.2.5 Size of the programs . . . 134
6.3 Integer linear programming models for PpCP . . . 134
6.3.1 MIP formulation for `PpCP` . . . 135
6.3.2 IP formulation for `PpCP` . . . 135
6.3.3 Size of the programs . . . 137
6.4 An exact solution method for RpCP . . . 139
6.4.1 Upper and lower bounds . . . 139
6.4.2 Exact solution method . . . 140
6.4.3 Experimental results . . . 142

7 Conclusion **157**
7.1 Context and contributions . . . 157
7.2 Open problems . . . 160
7.2.1 Theoretical open questions . . . 160
7.2.2 Practical open questions . . . 162
7.3 Production . . . 163
7.3.1 List of publications . . . 163
7.3.2 Conferences . . . 163

A List of Problems **165**

Bibliography **168**

LIST OF FIGURES

1 Example d'une instance avec une solution optimale différente pour `Min p-Center` et `RpCP` pour $p = 2$. 16

2.1 Distance to the nearest shelter when no fire occurs. 20
2.2 Evacuation distances in the scenario 1. 21
2.3 The operational graph associated with scenario 2 with shelters located on $\{3, 10\}$ 22
2.4 The operational graph associated with scenario 3 with shelters located on $\{3, 10\}$ 22
2.5 A path with 5 vertices and shelters on the extremities 27
2.6 An example of different optimal solutions for `Min p-Center` and `RpCP` with $p = 2$. 28
2.7 An example where, for $p = 5$, there is an optimal solution for `RpCP` and `PpCP` of size 4. 28

3.1 The gadget L_4 for an edge (u, v) 47
3.2 The subgraph $\widetilde{T}_{uv}$ for the edge (u, v) oriented from u to v and $\ell_{uv} > 1$. Continuous lines are used for T_{uv}'s edges while dashed lines correspond to edges outside T_{uv}. 50
3.3 The different graphs involved in the reduction. 58
3.4 Example of Transformation 1 60
3.5 The graph $\widetilde{H}_q$ obtained from G through H_q. $k_{ab} = k_{bc} = k_{bd} = 1$, $k_{ad} = k_{cb} = 3$ and $k_{ac} = 7$ 61
3.6 Gadget T^2_{ux} used in F for $(u, i), (i, x) \in E^{H_q}$; $z^7_{ux} = i$. 62
3.7 Gadget T^1_{xy} used in F for $(x, y) \in E^{H_q}$. 62

4.1 Segments induced by solution $\{1, 6, 9, 17\}$ on P_{17} 71

4.2 Evacuation radius induced by scenarios $s \in C$ where $C = \{1, 5, 8, 12\}$ on uniform P_{12}. 76
4.3 Evacuation radius induced by scenarios $s \in C$ where $C = \{1, 4, 8, 12\}$ on uniform P_{12}. 76
4.4 The unique balanced solution for PpCP with $p = 3$ on G 81
4.5 A non-balanced solution for PpCP with $p = 3$ on G 81
4.6 A case where $\frac{\mathbf{E}(C)}{r(C)} = Z + 1$. 83
4.7 Distance relations between vertices x, s and x_k used for Proposition 36 85
4.8 With general weights, an optimal MAC p-set can be a very bad PpCP solution. 87

5.1 Values λ_i^C, α_i^C, ϕ_i^C and β_i^C illustrated on a segment μ_i^C 100
5.2 Example graph on which we apply Algorithm 2 for $p = 3$ 103
5.3 Example graph G' on which we apply Algorithm 2 for $p = 3$ 104
5.4 Example of notations on an extended star. 105
5.5 Example of the notation of a caterpillar. 108
5.6 Applying Algorithm 5 on a caterpillar for k-RCP with $k = 5$, the red arc shows the relocation of the shelter from vertex 6 to vertex 7 at the beginning of iteration 7. 114

6.1 Processing time of each variant of the Exact Algorithm on some OR-Library instances 144
6.2 Processing time of EA3 on three different instances for p ranging from 2 to 32 147

LIST OF TABLES

1 Récapitulatif des principaux résultats pour le RpCP et le PpCP. . . . 20

2.1 Review 35

5.1 Description of the evolution of the parameters of the Algorithm 3 applied to G. 104

6.1 Number of variables and constraints for each presented formulation. 129
6.2 Number of variables and constraints for each presented formulation of RpCP. 134
6.3 Number of variables and constraints for each formulation introduced for PpCP. 137
6.4 Optimal solution values and bound values for OR-Library instances 143
6.5 Comparison of the standardized execution time of EA3 for different values of σ_1 146
6.6 Ratio between UB^1 and the optimal value of RpCP on OR-Library instances 148
6.7 Optimal solution values and bound values for unit subgrids 150
6.8 Optimal solution values and bound values for weighted subgrids . . 151
6.9 Ratio between UB^1 and optimal solution value of RpCP on unit subgrids 152
6.10 Ratio between UB^1 and optimal solution value of RpCP on weighted subgrids 152
6.11 Processing time of each variant of the Exact Algorithm on the unit subgrids 154
6.12 Processing time of each variant of the Exact Algorithm on the weighted subgrids 155

6.13 Best results obtained on the unsolved instances of unit subgrids . . 156

7.1 Hardness, approximation and non-approximation results. 160

CHAPTER 1

INTRODUCTION

1.1 Context and Motivations

Due to the effects of global warming, the increase in number and in severity of forest fires, or wild fires, is a matter of concern around the world [1]. New territories, previously free of such risk, become subject to fires [2]. Populations living in these areas and even first responders are not accustomed to such events and thus poorly prepared. As for populations culturally accustomed to facing this kind of challenges, they find themselves increasingly overwhelmed by exceptional events. A new term, "megafire", has appeared in the scientific literature to describe these wildfires of gigantic proportions and breaking free from the rules and observations previously established for smaller fires [3]. This trend seems set to normalize: each year, new records are broken by the wildfires in the two hemispheres of the globe.

These fires cause significant social and ecological losses. To give an idea, in 2018, the U.S. Fire Administration counted at least 3655 death, and estimated $25 billion (2018 USD) of loss [4]. For the 2019–20 Australian bushfire season, more than 400 direct and indirect human casualties were counted, more than a billion animals were killed, and more than 9000 buildings including 3500 houses were destroyed [5]. Given the high stakes involved, many scientific disciplines are mobilized in order to better understand these new phenomena. Numerous recent studies highlight the need to move on to new policies that attend to the complex and dynamic interrelations between social, economic, environmental, and political drivers [2]. Awareness campaigns must be developed over the risk of fires in all affected regions, so that individuals and collectives can prepare for this type of events. In addition, the scientific community develops new methods and tools to help limiting the impacts of fires on our societies and environment.

This joint PhD was carried out as part of the European project "Marie Sklodowska-Curie Research and Innovation Staff Exchange (RISE)" between Europe and Australia entitled "Geospatial based Environment for Optimization Systems Addressing Fire Emergencies", GEO-SAFE [6]. The GEO-SAFE project started in May 2016 and aims at proposing innovative methods to deal effectively with problems related to forest fires. The project tackles various issues related to fire management, consolidated under 4 different work packages. The first work package, *Stochastic Mapping*, concerns spatial statistical analysis for the development of a stochastic risk, vulnerability and hazard cartography of a region regarding the likelihood and impact of a wildfire. The second work package, *Innovative Models*, is primarily concerned with designing new models or improving existing ones in the following fields: fuel management, fire suppression and protection and large scale evacuation modeling. The third work package, *Robustness and Efficiency*, aims the development of robust models for wildfire management and the design of new algorithms for efficient and flexible responses for initial and extended attack in wildfire emergencies. This thesis is part of this work package. Finally, the fourth work package, *Implementation and Training Tools*, involves modeling and managing fire risk, evacuation models and protocols, and training scenarios for all those engaged in fire management, including the general public.

Natural disasters like forest fires require an emergency management. A characteristic of natural disaster emergencies is the difficult decision environment, public authorities facing tight timelines, constrained resources, huge amount of data to analyze, extreme uncertainty and multiple objectives that may conflict as well as strong constraints when dealing with security issues and lives saving [7]. Emergency management consists in a continuous process with four key components: mitigation & prevention; preparedness; response; and recovery [8]. The goal of mitigation and prevention is to decrease the need for emergency response. The goal of emergency preparedness is to ensure that a rapid, coordinated and effective response is possible when an emergency occurs. Preparedness involves an investment of time and commitment of resources but is irreplaceable when an emergency arises. When the disaster occurs, it is time to follow an established response plan – not the time to create one. The goal of an effective recovery is to restore the affected infrastructure as quickly as possible. It leads directly into the mitigation and prevention phase to minimize the risk of a recurrence or development of a similar emergency situation in the future.

In this context, we address the problem of locating shelters to mitigate the casualties caused by a fire. We propose new shelter location models as two-phase problems, where the first phase incorporates elements of prevention and preparedness, while the second phase corresponds to the response.

A shelter is a safe place that provides cover from the fire. It is typically a building equipped to protect people against flames, but it can also be an emergency gathering point. Public authorities have shown increasing interest for such kind of shelters [9]. Although the idea is not new, its effectiveness fundamentally depends on the location of the shelters: as the territory to be covered by one or more shelters can be arbitrarily large, the decision-maker must take into consideration the paths traveled and the amount of time needed for people scattered in the territory to reach a shelter. It is unfortunately common to find victims in their cars, trapped by the fire while evacuating [10]. In addition, the question of guaranteeing safe evacuation conditions for hikers and other forest users also arises. Unless public access to forests and natural parks is completely prohibited for long periods of the year, which can be culturally hard to accept, it is necessary to think about new solutions. In both cases, public infrastructures as fire shelters can reduce the evacuation time and thus the number of casualties. In addition, it allows to save resources (fire fighters, first responders) that would otherwise have to be employed to ensure these people's safety. As fire outbreaks and their impacts are by nature unpredictable, it can be difficult to choose shelter locations that remain suited to each fire configuration. Recent studies have highlighted the need to develop new models for the location of fire shelters [10],[11], which was confirmed in discussions with final users as part of the GEO-SAFE project.

The problems of determining the "best" location for a set of facilities in order to serve a set of demand points are generally referred as *Facility Location* problems [12]. These problems lie at the core of the *Location Science* research area in the field of Operations Research, and more precisely of Combinatorial Optimization. The meaning of "best" depends on the nature of the problem under study, in particular the constraints and of the optimality criteria considered. Stimulated by real-world problems, facility location has been applied in many areas with great success, for example in logistics, telecommunications, routing and transportation [13, 14]. In the case of an emergency service (ambulances, fire brigades, police stations, etc.), facility location models have been proposed to minimize response time. A whole section of Location Science is devoted to problems where the input data available to decision-makers are incomplete or non deterministic at the time of decision-making. Such models are said under *indeterminacy*. Indeterminacy can affect demand, or the availability of facilities service, or the infrastructure availability. Models under indeterminacy are commonly used for natural disaster situations, such as floods, earthquakes, or forest fires. Indeed, a natural disaster, like forest fires, is an unforeseeable event, that produces sudden demand (areas are affected only once the hazard occurs) for services (e.g. emergency care, evacuation) and potentially various disruptions on the transport network. We present a

new shelter location model within this setup in what follows.

We want to locate a limited number of shelters on a territory where the general objective is to minimize the risk of having people trapped by the flames, i.e., to maximize their chance to reach a safe shelter before the fire gets too close. We consider territories with weak density of residents and forest users. We represent the territory by an adjacency graph. Each vertex corresponds to an area and two vertices are connected by an edge if the related areas are adjacent. The fire starts on an area, the corresponding vertex is called the *source* vertex. We call *scenario* a description of a potential future situation; we will define it more formally later on. In a real case, the fire can spread over very large areas and the impacted zone grows dynamically. However, it might be relevant to focus on a relatively short period after ignition or after the alert, seen as the time required for all people present in the area to reach a shelter. This motivates us to consider only scenarios where the fire is restricted to its source vertex. We assume an efficient early warning system with clear messages to evacuate, and that people are safe after reaching the shelter; this hypothesis is relevant depending on the exact nature and design of the shelters. As the fire is likely to spread over the whole territory, everybody in the territory is expected to evacuate to a shelter once the warning is triggered. In our context, shelters are generally build with relatively large capacity [9], thus we are not concerned with shelter capacity. In theory, we should consider two different networks in order to distinguish between the paths people can follow, and the fire spread. However, as we consider only scenarios where the fire is restricted to one area, we can simplify our problem by considering only the network along which people can move and represented by a graph.

Our problem can then be described as a particular case of *location-allocation problems*. A solution describes the location of shelters among the vertices of the graph, which corresponds to a subset of vertices. In addition, a solution also involves, for each scenario and for the people in a given area, a possible allocation of shelters to reach. The location can be addressed during the preparedness phase and may involve significant time consuming computation. The allocation however is mainly addressed during the response phase, which usually requires efficient and simple processes. More precisely, these allocations correspond to *evacuation paths* followed by the people evacuating the territory. We call *evacuation strategy* the actions that are expected to be taken by the people evacuating a territory once a fire hazard occurs.

The main contribution of this thesis is to propose a new evacuation strategy, the *Under Pressure* evacuation strategy, which describes evacuation decisions made under pressure. Its main originality is as follows: if no shelter is located on the source vertex, then we consider there is an evacuation path leaving the source

vertex towards every adjacent vertex. The justification is twofold: the fire in the source vertex restricts the internal mobility; the immediate threat of the fire may affect in different ways the behavior of the people in this area. The evacuation path of other vertices of the graph is a shortest path between the vertex and a nearest reachable shelter. We call *evacuation distance* of a vertex the maximum length of one of its evacuation paths. For a real case implementation, this could be supported by simple informative signs deployed on site, or marked on hiking maps, indicating the direction and distance of the closest accessible shelters. This evacuation strategy induces particular evacuation distances which render our model specific. To our knowledge, it is the first attempt to model the evacuation strategy in such manner for a shelter location problem. We introduce two problems with the Under Pressure evacuation strategy, which differ by the indeterminacy environment considered. We present our problems as variants under indeterminacy of the deterministic `Min p-Center` problem [15]: the `Min p-Center` problem is a minimax facility location problem that consists in locating at most p identical facilities, on a graph to minimize the maximum distance between demand vertices and a closest facility. This model is however insufficient to address the indeterminacy induced by the fire and the evacuees' behavior.

With the Under Pressure evacuation strategy, the value of a solution for a given scenario is given by the maximum evacuation distance in that scenario, called the *evacuation radius*. We propose to study two categories of indeterminacy environment: risk and uncertainty [16]. In risk situations, there are uncertain parameters whose values follow some probability distributions that are known by the decision maker; this setup corresponds then to stochastic or probabilistic optimization. The *Probabilistic p-Center Under Pressure* problem is the problem of locating at most p shelters such that the expected evacuation radius, given a probability distribution on scenarios, is minimized. In uncertainty situations, parameters are uncertain and no information about probabilities is used; this setup corresponds then to robust optimization. The *Robust p-Center Under Pressure* problem is the problem of locating at most p shelters such that the maximum evacuation radius, given uncertain scenarios, is minimized.

To summarize, we list the strong assumptions in our problems, which has been discussed with final users as part of the GEO-SAFE project. We would like to thank particularly the Pau Costa Foundation [17], the Fire Organisation of Andalucia (INFOCA) [18], the Fire Organisation of Corsica (SDIS2B) [19], and the Country Fire Authority of Victoria, Australia (CFA) [20].

- The territory, with weak density of populations, is represented by an adjacency graph.
- Each vertex corresponds to an area.

- The fire is restricted to one vertex during the short period we consider. We consider only scenarios with one fire outbreak.
- The fire is likely to spread over the whole territory and for this reason, everybody in the territory is expected to evacuate to a shelter.
- Shelters protect from the fire and have large capacity.
- An efficient early warning system exists.
- Simple guidance signs are deployed on site, or marked on hiking maps, indicating the direction and distance of the closest accessible shelters.
- Fire can start in an area where a shelter is located. In this case, we assume that the shelter location in the area and the local layout guarantee easy accessibility [10] during the phase just after the fire outbreak for the people in the immediate surrounding. Given the risk of fire spread, this shelter is however inaccessible to people in other areas.
- The evacuation decisions made by the evacuees correspond to the Under Pressure evacuation strategy.

1.2 Basic Notations

In this section we introduce basic notations and notions needed for the understanding of our work.

Graphs

These notations are inspired from [21] by Korte and Vygen.

Undirected graph An undirected graph G is a couple (V, E) where $V = \{1, \ldots, n\}$ is the set of vertices and $E = \{X \subseteq V : |X| = 2\}$ the set of edges, with $|E| = m$. We denote (i, j) the edge joining vertices i and j, which are also called the endpoints of (i, j). An edge whose endpoints coincide is called a loop. Two edges with identical endpoints are said parallels. A graph without parallel edges nor loop is said simple. We say G is edge-weighted when a natural number $l_{i,j}$, called the length or the weight of the edge (i, j), is associated to each edge (i, j). We will refer as the *uniform case* the case where all edge lengths are equal. G is said vertex-weighted graph when a natural number, called the demand or the weight of the edge (i, j), is associated to each vertex $v \in V$. In this thesis, undirected graphs are exclusively simple graphs, therefore we do not specify it each time.

Unless specified otherwise, we consider non-weighted graphs. In addition, if i and j are joined by an edge, then i and j are called adjacent, and i is the neighbor of j (and vice versa). The degree of a vertex i is the number of vertices adjacent to i. A vertex of degree 1 is called a *pending* vertex.

Directed graph or Digraph A directed graph or digraph G is a couple (V, E) where $V = \{1, \ldots, n\}$ is the set of vertices and $E = \{(i, j) \in V \times V : i \neq j\}$ the set of directed edges.

Mixed graph A mixed graph is a graph with both directed and non-directed edges. If (i, j) is a directed edge, we say (i, j) leaves i and enters j.

When no ambiguity occurs, we will use similar notations for directed, undirected and mixed graphs. We will just indicate directed edges an denote them with an arrow in the related drawing. We use mainly undirected and mixed graphs. All non-directed notions in graphs also apply to mixed graphs by considering the *non-directed version* of the mixed graph obtained by replacing directed edges by non-directed ones, and then replacing multiple parallel edges by a simple edge. Similarly, all directed notions apply to mixed graphs since a mixed graph can be seen as a digraph with non-directed edges replaced by two directed edges in opposite directions.

Subgraph and partial graph A subgraph of a mixed graph $G = (V, E)$ is a mixed graph $H = (X, Y)$ where $X \subseteq V$ and $Y = \{(i, j) : (i, j) \in E \text{ and } i, j \in X\}$. We also write $H = G[X]$ and say H is the subgraph of G induced by X. A partial graph of G is a mixed graph (V, E') with $E' \subseteq E$ obtained from G by deleting zero or some edges. A partial subgraph of G is a partial graph of a subgraph of G. For some $U \subset V$, we denote $G \setminus U$ the graph $G[V \setminus U]$. For some $v \in V$, we denote $G - v$ the induced subgraph $G \setminus \{v\}$. If H is a partial subgraph of G, we say G contains H.

Path An undirected path is an undirected graph $P = (\{v_1, \ldots, v_{k+1}\}, \{(v_1, v_2), \ldots, (v_k, v_{k+1})\}$ such that $v_i \neq v_j$ for $1 \leq i \leq j \leq k+1$. G is also called a path from v_1 to v_{k+1}, or between v_1 and v_{k+1}. v_1 and v_{k+1} are the extremities of P. When speaking about directed paths in a mixed graph, paths are meant to respect the edge orientations. The length of a path is then the sum of the lengths of its edges. In a general graph G, if G contains as partial subgraph a path P, we say P is a path in G.

Connected components Let $G = (V, E)$ be some undirected graph. G is connected if there is a path between all pairs of vertices, otherwise G is disconnected. Sometimes we identify the connected components with the vertex sets inducing them. A vertex v is called an *articulation vertex* if the induced subgraph $G - v$ has more connected components than G. We denote by $\mathcal{A}(G)$ the set of articulation vertices of G. We call *articulation components* of G the set of connected components of $G - a$ with $a \in \mathcal{A}(G)$. A graph is 2-connected if it has no articulation point; in this case there is no articulation component.

Acyclic graphs A cycle is a graph $G = (\{v_1, \ldots, v_{k+1}\}, \{(v_1, v_2), (v_2, v_3), \ldots, (v_k, v_{k+1})\}$ such that $v_1 = v_{k+1}$ and $v_i \neq v_j$ for $1 \leq i \leq j \leq k$. A *forest* is a graph without any cycle as a partial subgraph. A connected forest is a *tree*. A *star* is a tree where at most one vertex is not a pending vertex. A *caterpillar* is a tree in which all the vertices are within a central path or adjacent to it. It can also be described as a tree in which all vertices that are not pending form a path.

Planar graph A planar graph is a graph that can be embedded in the plane, i.e., it can be drawn on the plane in such a way that its edges intersect only at their endpoints.

Grid, subgrid and partial subgrid A $n \times m$ grid is the planar graph $G = (V, E)$ with vertex set $V = \{v_{i,j}, i \in \{0, \ldots n-1\}, j \in \{0, \ldots, m-1\}\}$ and $(v_{i,j}, v_{k,l}) \in E$ if and only if $|i-k|+|j-l| = 1$. A (partial) subgrid is a (partial) subgraph of a grid.

Let $G = (V, E)$ a mixed graph, and $C \subseteq V$. We denote:

- $d(i, j)$ the distance between i and j i.e., the length of a shortest path from i to j. If no such path exists, $d(i, j) = +\infty$. When speaking about distances in a mixed graph, paths are meant to respect the edge orientations and thus, the matrix of distances is not symmetric anymore.
- $d(i, C) = \min_{j \in C} d(i, j)$.
- $r(C) = \max_{v \in V} d(v, C)$ is called the *radius* of C.
- $\Gamma(i) \subset V$ is called the set of neighbors of vertex i.
- $deg(i) = |\Gamma(i)|$ the degree of vertex i.

Algorithmic and Mathematical Notions

Decision problems and optimization problems A decision problem is a problem that can be posed as a yes-no question on the input values."An optimization problem is a problem of determining a best solution among the set of solutions satisfying some properties imposed by the very definition of the problem" [22]. An optimization problem I can be expressed as the following mathematical program:

$$\begin{cases} opt f(\vec{x}) \\ \quad \vec{x} \in \mathtt{C_I} \end{cases}$$

where $\vec{x}$ is a vector describing a solution, $\mathtt{C_I}$ is the constraints of the problem, f is the objective function and $opt \in \{\min, \max\}$. An optimization problem has always a decision version. Let $\Pi = (I, z)$ be the problem of deciding whether there is a $\vec{y} \in \mathtt{C_I}$, such that $f(\vec{y}) \leq z$ if $opt = \min$ (respectively $f(\vec{y}) \geq z$ if $opt = \max$). Π is the decision version of optimization problem I. We use decision versions of an optimization problem when discussing its complexity.

NP-completeness and NP-hardness Roughly speaking, a decision problem Π is in NP if a certificate that an instance is a yes-instance can be checked in polynomial time. A decision problem Π is NP-complete if Π is in NP, and all other problems in NP polynomially reduce [23] to Π. An optimization problem is said in NP if its decision version is in NP. An optimization problem is called NP-hard if and only if its decision version is NP-complete [22].

Approximation An α-approximation algorithm for an optimization problem is a polynomial-time algorithm that, for all instances of the problem, produces a feasible solution whose value is within a factor of α of the optimal value [24]. Therefore the approximation ratio of an algorithm is always larger than 1 for minimization problems (smaller than 1 for maximization problems). The closer to 1, the better the algorithm. For example, a 2-approximation algorithm for a minimization problem is a polynomial-time algorithm that always returns a solution whose value is at most twice the optimal value.

Binary search Binary search refers to successively halving a finite list of numbers using a median element of the list and discarding either the lower or the upper half in each step until its size is smaller or equal than a predetermined positive natural number, by default 1 in this thesis.

Argument of the minimum For a real-valued function f with domain S, $\underset{x \in S}{\operatorname{argmin}}\ f(x)$ is the set of elements in S that achieve the global minimum in S:

$$\underset{x \in S}{\operatorname{argmin}}\ f(x) = \{x \in S : f(x) = \min_{y \in S} f(y)\}$$

Next we present some classical optimization problems relevant to our thesis, defined on an undirected graph $G = (V, E)$.

Usual problems

The p-Center problem denoted `Min p-Center` [15]

`Min p-Center`
Instance: A graph $G = (V, E)$, an integer p
Feasible solutions: A p-set i.e., a set $C \subseteq V$ of size at most p
Objective: Find a p-set C of minimum radius $r(C)$.

Note that an optimal solution of size p always exists.

The Minimum Vertex Cover problem denoted `Min Vertex Cover` [25]

`Min Vertex Cover`
Instance: A graph $G = (V, E)$
Feasible solutions: A vertex cover i.e., a set $U \subseteq V$ such that every edge of E is incident to at least one vertex of U
Objective: Find a vertex cover of minimum size.

The minimum size of a vertex cover G is denoted $\tau(G)$.

The Minimum Dominating Set problem denoted `Min Dominating Set` [25]

`Min Dominating Set`
Instance: A graph $G = (V, E)$
Feasible solutions: A dominating set i.e., a set $D \subseteq V$ such that every vertex of $V \setminus D$ is adjacent to a vertex of D.
Objective: Find a dominating set of minimum size.

The minimum size of a dominating set in G is called the domination number of G and is denoted $\gamma(G)$.

The Minimum Set Covering problem denoted `Min Set Cover` [21]

`Min Set Cover`
Instance: A set system $(\Omega, \mathcal{Z})$ with $\cup_{Z\in\mathcal{Z}} Z = \Omega$
Feasible solutions: A set cover i.e., a set $\mathcal{Y} \subseteq \mathcal{Z}$ such that $\cup_{Y\in\mathcal{Y}} Y = \Omega$
Objective: Find a set cover of minimum size.

The Minimum radius Dominating Set problem denoted `Min r-Dominating Set` [26]

`Min r-Dominating Set`
Instance: A graph $G = (V, E)$, an integer r
Feasible solutions: A set $C \subseteq V$ such that $r(C) \leq r$
Objective: Find a feasible solution of minimum size.

`Min r-Dominating Set` is a generalization of `Min Dominating Set` and a specific case of `Min Set Cover`. We define it separately as we will frequently refer to it in this form.

1.3 Organization of the thesis

This document is organized as follows.

In Chapter 2, we describe thoroughly our model for shelter location in a forest fire context. We present our modeling hypothesis for representing the territory, the shelters, the uncertainty of fire outbreaks, the evacuation strategy and the objective functions considered. We introduce the *Robust p-Center Under Pressure* problem (`RpCP`) and the *Probabilistic p-Center Under Pressure* problem (`PpCP`), as well as the set of feasible solutions for these problems. Finally, we compare our models to some relevant related works in the literature.

In Chapter 3, we analyze the hardness of `RpCP` and `PpCP`. We present different reductions from classic deterministic problems like `Min Vertex Cover` and `Min Dominating Set` to decision versions of `RpCP`. We also present first non-approximation results for `PpCP`. We take care of establishing hardness results on classes of graphs with practical significance, like planar graphs or subgrids. Note that, the more restrictive the class of graph, the stronger the hardness result.

In Chapter 4, we present polynomial approaches to generate exact or approximation results for `PpCP`. We propose an explicit solution on paths in the uniform case by using auxiliary problems. The proof is surprisingly non-trivial. Then we

propose a constant-ratio approximation algorithm for PpCP on general graphs with bounded degrees and bounded length of the edges. Here again, we will use auxiliary problems that reveal to be interesting on their own.

In Chapter 5, we propose polynomial algorithms for RpCP on different classes of acyclic edge-weighted graphs. Using structural properties introduced in the previous chapter, we will present different methods based on a binary search. We propose refined polynomial algorithms for paths, extended stars, caterpillars and trees.

In Chapter 6, we present different mathematical methods to compute exact solutions of RpCP and PpCP. These methods are inspired by the main existing integer linear programs for Min p-Center, but require extended generalization to represent the undeterminacy and take into account the Under Pressure evacuation strategy. For RpCP, we propose an even more efficient algorithm, tractable on large instances as shown by our experimental results.

Finally, in Chapter 7, we summarize the contributions of this thesis, and we indicate several lines of future research.

In Appendix A, the reader can find a list of the problems considered in this thesis.

CHAPTER 2

PRESENTATION OF THE PROBLEMS

Abstract

In this chapter we introduce our model for shelter location in a forest fire context and specifically a new original evacuation strategy called Under Pressure. We then define the two main problems studied in this thesis, namely the *Probabilistic p-Center under Pressure* problem and the *Robust p-Center under Pressure* problem. We determine the set of feasible solutions for these problems. Finally, we emphasize the differences of these new problems with existing models in the literature.

Contents

2.1 The *Under Pressure* model 16

2.1.1 Modeling the territory 17

2.1.2 Scenarios 18

2.1.3 The Under Pressure evacuation strategy 18

2.1.4 Objective functions 23

2.1.5 Feasibility conditions 25

2.1.6 Examples and comparison with *p-Center* 27

2.2 State of the Art 29

2.2.1 The *p*-Center Problem 29

2.2.2 Models under indeterminacy 34

Introduction

The problems of determining the "best" location for a set of facilities in order to serve a set of demand points are generally referred to as *Facility Location* problems. These problems lie at the core of the *Location Science* research area [12]. The meaning of "best" depends on the nature of the problem under study, namely in terms of the constraints and of the optimality criteria considered. Facility Location problems have been widely studied for emergency applications, in particular the `Min p-Center` problem. `Min p-Center` is particularly suited to decision makers concerned with the reliability of the location for every user and who want to consider a criterion focusing on users who are the poorest served (for an extended description of the state of the art, see Subsection 2.2.1). A main objective in our context is to avoid any death toll, which favors `Min p-Center` for a shelter location decision-support model. `Min p-Center` is however not sufficient to handle the indeterminacy induced by the fire hazard. To this end, we propose a new approach, which we present in Section 2.1. Then, in Section 2.2, we compare our approach to the existing state of the art.

Our model was first presented in [27] for the probabilist case, and in [28] for the robust case.

2.1 The *Under Pressure* model

When a fire occurs, all the people within the territory must head for a safe place, which we will refer to as a shelter. Our objective is to determine the best location for a set of shelters in order to minimize the distance traveled by the people once a fire occurs. We will model the territory as a graph like described in Section 2.1.1, such that shelters locations correspond to a subset of vertices. We will model the indeterminacy relative to fire ignition using scenarios as described in Section 2.1.2.

Our problem is a particular case of location-allocation problems in two phases: a solution describes the a priori location of shelters among the vertices of the graph. This task must be undertaken during the preparedness phase. In addition, our solution must describe, for each scenario, an allocation of areas to shelters. This allocation corresponds to evacuation paths that must be taken by the people during the response phase. The allocation decision is made on the basis of an evacuation strategy. In Section 2.1.3, we describe a new original evacuation strategy called *Under Pressure* which takes into account some behaviors that may occur in fire emergency situations.

The examination of the *Under Pressure* evacuation strategy induces new evacuation distances which renders the problem of choosing a best location for shelters specific compared to the literature and introduces some additional complexity. This new evacuation strategy, applied to a version of `Min p-Center` under indeterminacy, generates new problems relevant for shelter location under wildfire context. This thesis mainly focuses on two of these problems, presented in Section 2.1.4: the *Robust p-Center Under Pressure* and the *Probabilistic p-Center Under Pressure*. We also present some first results on the differences between these problems and `Min p-Center`.

Finally, in Section 2.1.5 we present the necessary and sufficient conditions for a solution to be feasible for the two previously defined problem. We also present in Section 2.1.6 some first results on the differences between these new problems and `Min p-Center`.

2.1.1 Modeling the territory

In our model, the territory, typically with low density habitat, is represented by an edge-weighted graph $G = (V, E)$ with $V = \{1, \ldots, n\}$ the set of vertices, and E the set of weighted edges. Each vertex corresponds to an area or zone; the areas are defined based on geographical and spatial criteria such that the center of the area is a candidate location for a shelter. In particular, natural barriers like a river or a cliff will always be placed on boundaries between zones. Typical examples of zones could be a village and its suburb areas or, in case of very sparse habitat, a homogeneous area with easy circulation inside. The size of zones may also vary. We assume that, in absence of fire, it is easy to move inside an area. It is not the intention of this thesis to determine the areas. If adjacent areas correspond to a fragmentation of a large homogeneous territory without clear natural boundaries, then the related edges are just a discrete model of a continuous reality. In other cases however, edges represent an accurate representation of the real transport network environment. Since vertices represent areas, possibly large, the length of an edge can be seen as a distance or a traveling time between adjacent vertices and can be measured between median points in the two related areas or as a maximum distance; these choices has no incidence on the model. According to our model, shelters must be placed on vertices. Hence, shelter locations correspond to a subset of vertices, denoted C in the following. Given the analogy to `Min p-Center`, we refer to the vertices of C as shelters or centers without distinction.

Next we present how we take into account in our model the indeterminacy induced by the fire hazard.

2.1.2 Scenarios

A classical approach to model disaster risk is to use scenarios [29]. A set of scenarios is a description of the states of the system after a hazard occurs. In real cases applications, the definition of scenarios relies on a preliminary analysis and synthesis of the effects of a vast set of parameters. After which, a finite set of scenarios should be defined based on major uncertainty factors of the decision-making environment. It follows that the quality of the solution produced fundamentally depends on the scenario generation process.

We denote S the set of scenarios. A scenario is associated with a specific fire outbreak in the territory represented by the edge-weighted graph $G = (V, E)$. There is a potentially large number of scenarios (several simultaneous fire outbreaks in different areas, fire spreading more quickly in one direction, fire jumping from one area to another through ambers, etc). We restrict ourselves to a single fire outbreak event with the fire spread limited to its native area. Consequently, we restrict S to the set of scenarios corresponding to a single vertex on fire, and we refer to the scenario associated with a fire outbreak on vertex $s \in V$ as scenario $s \in S$. So $S = V$. This restriction is motivated by our primary focus on a relatively short time period after an outbreak, which assumes an efficient early warning system. In this case, everybody can escape to a shelter before the fire spreads to adjacent areas. In this thesis, we refer to the vertex on fire also as the source vertex: in scenario s, vertex s is the source vertex.

The *operational graph* associated with the scenario s, denoted by $G^s = (V, E^s)$, is a mixed graph obtained from G as follows: all edges $e \in E$ are kept in E^s except the edges (s, v) incident to s: these edges are replaced by a unique directed edge (s, v). Consequently, paths are directed in G^s and vertex s is no longer accessible from another vertex. For a scenario s and for any $i, j \in V$, we denote by $d^s(i, j)$ the length of a directed shortest path from i to j in G^s. In addition, for $C \subseteq V$, we denote $d^s(i, C) = \max_{j \in C} d^s(i, j)$.

Note that, for all $j \in V \setminus \{s\}$, there is no path from j to s, thus $d^s(j, s) = +\infty$. In addition, if $i \neq s$ and $j \neq s$, $d^s(i, j) = d^s(j, i)$.

2.1.3 The Under Pressure evacuation strategy

We have seen that a solution does not only describes the location of shelters, it also specifies, for each scenario and for the people in a given area, a shelter to reach on the basis of an evacuation strategy. The evacuation strategy describes the actions that are expected to be taken by the people evacuating a territory once a fire hazard occurs. A good shelter location model for wild fire context must try

to take into account the actions people on site are likely to undertake, in order to locate shelters in such way that it serves the population.

The decisions of where to locate shelters and how the evacuation planning is made are equally important as both affect the time to evacuate the region affected by the disaster. Addressing these two problems separately may lead to suboptimal results. There are two classical evacuation planning models [30]. The first one considers that people will head for the nearest safe shelter, driven by the intuitive reaction of any individual looking for the nearest accessible shelter, which is not necessarily in the interest of the whole system. To our knowledge, all the problems based on this approach model the evacuation path as a shortest path in the operational graph. The second approach involves the assignment of evacuation routes to the travelers, for example in traffic assignment models [30]. This approach requires rapid customized communication with all the people to be evacuated. The limit of these approaches and the need for new shelter location models have been outlined by different works on the topic [11, 30, 10] and confirmed by final users as part of the GEO-SAFE project.

We propose an alternative approach, called the *Under Pressure* evacuation strategy and denoted the UP evacuation strategy. We consider that rapid customized communication with all the people to be evacuated is not always possible, for example when people on site are daily visitors. However, as shelters have been built well ahead of time, we can assume signals or early warnings are in place such that people know their exact positions. On top of the idea that any individual intuitively looks for the nearest accessible shelter, we propose specific implications on the actual paths traveled by the people once they evacuate.

Consider a territory modeled with an edge-weighted graph $G = (V, E)$, scenarios S, and a set of shelters $C \subset V$. When a fire occurs, all the people within the territory must head for a safe place. An *evacuation path* of v in scenario s is a path in G^s that individuals on v may follow to reach an accessible shelter in C. Three cases emerge, depending on the whereabout of the area v :

1. For people in area s, two cases have to be considered.
   (a) If a shelter is located in area s, we assume that all the persons present in that area can safely reach this shelter. To support this hypothesis, we assume that the shelter location in the area and the local layout guarantee easy accessibility [10] during the phase just after outbreak. It is reasonable to assume that clear signs direct people accurately on and around the shelter. However, for people outside the area, attempting to reach the shelter could be dangerous. For this reason s is not accessible from other vertices in the operational graph G^s.

(b) If there are no shelters in area s, we assume that people in this area will first flee in any other direction to reach an adjacent area j, then they will evacuate to the nearest shelters accessible from j in G^s. Without strong evidence (like a shelter in the area), people, especially under pressure, may react in very diverse ways: choosing for instance to run in the opposite direction of the fire, or in the orthogonal direction of the wind or in the most accessible direction, regardless of the best direction for the nearest shelter. Once evacuees reach a safer adjacent area, they will be able to follow the shortest path to a shelter, i.e an evacuation path of s is a path $\{s, j, i_1, \ldots, i_k\}$ where $j \in \Gamma(s)$,$i_k \in C$ and the path $\{j, i_1, \ldots, i_k\}$ is a shortest path between j and i_k.

2. For people who are not in area s, that is any area $v \neq s$, we assume that going to the nearest accessible shelter is safe as long as the area on fire s is avoided. It is reasonable that such strategy will be adopted in practice. Then, the evacuation path of v in scenario s is the shortest path between v and a shelter in G^s.

To ensure an acceptable level of risk, we must consider the worst case scenario. The *evacuation distance* of a vertex v in scenario s is the distance (maximum distance in the case $v = s$ and $v \notin C$) traveled by the people from v to a nearest reachable shelter i.e., the maximum length of an evacuation path of v in scenario s. The evacuation distance of a vertex v in scenario s is then given by:

$$r^s(C, v) = \begin{cases} 0 & \text{if } v = s \text{ and } v \in C \\ \max\limits_{j \in \Gamma(v)} \{l_{v,j} + d^s(j, C)\} & \text{if } v = s \text{ and } v \notin C \\ d^s(v, C) & \text{if } v \neq s \end{cases} \tag{2.1}$$

Note that, for a scenario s, if there is no path from v to a shelter of C, we get $r^s(C, v) = +\infty$.

Example. On a path: We illustrate these definitions in Figures 2.1 and 2.2 on a path of nine vertices in the uniform case. Let $C = \{0, 5, 8\}$ be the set of shelters represented by pentagons. The value of $d(i, C)$ is given under each vertex i.

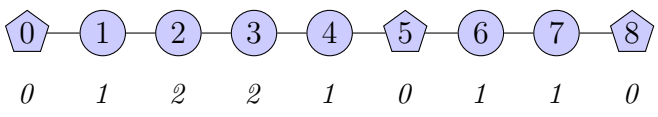


Figure 2.1: Distance to the nearest shelter when no fire occurs.

In Figure 2.2, we consider the operational graph G^1 associated to scenario 1, where the two undirected edges (i, j) and (j, i) are represented by a unique continuous line. The label under each vertex $i \in V$ is the value of $d^1(i, C), i = \{0, \ldots, 8\}$. It is underlined when it differs from $d(i, C)$. $d^1(1, C)$ and $d^1(2, C)$ differs from $d(1, C)$ and $d(2, C)$ respectively since the path to shelter 0 is no more operational. For vertex 1, two evacuation paths exist: $\{1, 0\}$ and $\{1, 2, 3, 4, 5\}$. As we can't know in advance which evacuation path will be used given the local constraints for the evacuees, we consider the worst case. Then $r^1(C, 1) = \max_{j \in \Gamma(1)}(l_{1,j} + r^1(C, j)) = 1 + \max\{r^1(C, 0), r^1(C, 2)\} = 1 + \max\{d^1(0, 0), d^1(2, 5)\} = 1 + \max\{0, 3\} = 4$.

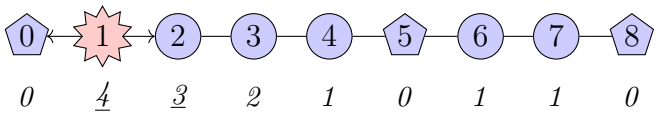


Figure 2.2: Evacuation distances in the scenario 1.

♣

Example. On a subgrid: In Figure 2.3, we consider a graph $G = (V, E)$ of 14 vertices. The figure illustrates the operational graph G^2, with shelters located on $C = \{3, 10\}$. Each edge is labeled by its length. When a fire occurs on vertex 2 (scenario 2), the evacuation paths of each vertex i corresponds to shortest paths from i to its nearest shelters, except for the following vertices:

- vertex 1: in G, the nearest shelter to vertex 1 is on 3 at a distance of 3, using the shortest path $\{1, 2, 3\}$. However, in scenario 2, the length of the shortest path from vertex 1 to vertex 3 is 22, using the shortest path $\{1, 6, 7, 8, 3\}$. Instead, the shelter in 10 is at a distance of 8 from vertex 1, using the shortest path $\{1, 6, 10\}$. Thus node 1 is evacuated to node 10 and $r^2(C, 1) = 8$.

- vertex 7: in G, the nearest shelter to vertex 7 is on 3 at a distance of 5, using the shortest path $\{7, 2, 3\}$. However, in scenario 2, the shortest path length from vertex 7 to vertex 3 is 11, using the shortest path $7, 8, 3$. Instead, the shelter in 10 is at a distance of 10 from vertex 7, using the shortest path $\{7, 11, 10\}$. Thus node 7 is evacuated to node 10 and $r^2(C, 7) = 10$.

- vertex 2: in G, the nearest shelter to vertex 2 is on its neighbor 3 at a distance of 2. However, in scenario 2, there are three evacuation paths originating from vertex 2:

  - the evacuation path passing through the neighbor vertex 1, at a distance of 1. From there the nearest shelter 10 is at a distance of 8. Thus the overall length of this evacuation path is $1 + 8 = 9$.

- the evacuation path passing through the neighbor vertex 7, at a distance of 3. From there the nearest shelter 10 is at a distance of 10. Thus the overall length of this evacuation path is 3 + 10 = 13.
- the evacuation path passing through the neighbor vertex 3, at a distance of 2. A shelter is located there, thus the overall length of this evacuation path is 2.

As any of these three evacuation paths is potentially taken by people evacuating vertex 2, the evacuation distance of vertex 2 is given by the longest of these evacuation paths. In this case, $r^2(C, 2) = 13$.

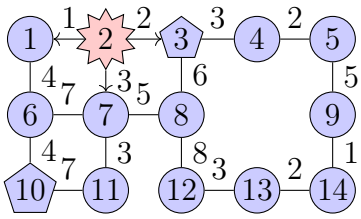


Figure 2.3: The operational graph associated with scenario 2 with shelters located on $\{3, 10\}$

In scenario 3, the fire occurs on a vertex with a shelter. The corresponding operational graph is illustrated in Figure 2.4. In this case, $r^3(C, 3) = 0$. All other evacuation distances correspond to the length of a shortest path from the originating vertex to shelter 10 in G^3. For example, the evacuation distance of vertex 4 is given by the length of the path $\{4, 5, 9, 14, 13, 12, 8, 7, 11, 10\}$, for a total distance of 36.

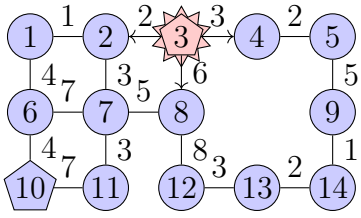


Figure 2.4: The operational graph associated with scenario 3 with shelters located on $\{3, 10\}$

♣

This characterization of the evacuation distances renders our model specific. In Section 2.2, we compare the UP approach to other evacuation strategies in the literature. To our knowledge, it is the first attempt to model the evacuation strategy in such manner for a shelter location problem.

2.1.4 Objective functions

For a scenario s, the quality of a shelter location is determined on the basis of the longest evacuation distance one may travel in G^s. Considering the UP evacuation strategy, we define the *evacuation radius* of a set $C \subset V$ for scenario s as follows:

$$r^s(C) = \max_{x \in V} r^s(C, x) \tag{2.2}$$

If the distance-matrix of G^s is known, $r^s(C)$ can be computed in $O(n^2)$.

Note that C induces potentially different evacuation radius for each scenario $s \in V$. The evaluation of C over all the set of scenarios depends on the considered environmental indeterminacy. We consider two categories of environmental indeterminacy: risk and uncertainty [16], [31]. In the case of risk, a probability distribution is known for the scenarios; this setup corresponds then to stochastic or probabilistic optimization. In the case of uncertainty, no probability is known; this setup corresponds then to robust optimization. We propose to study the two forms of indeterminacy with the UP evacuation strategy.

2.1.4.1 Robust optimization approach

The robust approach is suitable if no information is known about the probability of scenarios, or if it is critical to guarantee evacuation distances even in the worst case, independently of its likelihood. Minimax is a classical objective function in robustness approach [31]. Under a minimax objective function, the optimal decision is that for which the worst case performance of a solution over all scenarios is minimized. The minimax criterion applied to our context corresponds to the minimization of the maximum evacuation radius across all scenarios. Consequently, for a set $C \subset V$, we define the objective function $\mathbf{R}$ as follows:

$$\mathbf{R}(C) = \max_{s \in V} r^s(C) \tag{2.3}$$

$\mathbf{R}(C)$, called the *robust radius* of C, is the maximum value of the evacuation radius of C over all the scenarios. Hence, we can define a new problem.

Robust p-Center Under Pressure (`RpCP`)

Instance:	An edge-weighted graph $G = (V, E)$, a set of scenarios S, an integer p
Feasible solutions:	A p-set $C \subseteq V$ with finite $\mathbf{R}(C)$
Objective:	Find a solution of minimum robust radius if it exists.

Note that, if an optimal solution exists, an optimal solution of size p always exists. In addition, `RpCP` is closely related to another optimization problem which will be useful in our study in the following chapters.

`k-RCP`

Instance:	An edge-weighted graph $G = (V, E)$, a set of scenarios S, an integer k
Feasible solutions:	A set $C \subseteq V$ of robust radius $\mathbf{R}(C) \leq k$.
Objective:	Find a feasible solution of minimum size.

Consider an instance (G, S, p) of `RpCP`. We can propose an exact solution method using `k-RCP` coupled to a binary search as follows: Let D be the ordered set of candidate values for the value of `RpCP` on G i.e., $D = \{d^s(x, y) : x, y, s \in V\} \cup \{l_{sx} + d^s(x, y) : y, s \in V, x \in \Gamma(s)\}$. If the optimal value of `k-RCP` on (G, S, k) is greater than p, then the optimal value of `RpCP` is greater than k. Else, the optimal value is less or equal to k. We continue the binary search on a restricted subset of D to find the minimum radius k^* for which the optimal value of `k-RCP` is less or equal to p.

In this thesis, as we consider exclusively $S = V$, we will refer to an instance (G, S, p) of `RpCP` as (G, p). Similarly, we will refer to an instance (G, S, k) of `k-RCP` as (G, k)

2.1.4.2 Probabilistic optimization approach

The probabilistic approach is suitable when it is possible to assign probabilities to the scenarios. A common goal is then to optimize the expected value of the evacuation radius. We denote by π_s the probability of scenario s and by $\Pi = (\pi_s)_{s \in S}$ the vector of scenario probabilities.

For a set $C \subset V$, we define the objective function $\mathbf{E}$ as follows:

$$\mathbf{E}(C) = \sum_{s \in V} \pi_s \times r^s(C) \tag{2.4}$$

$\mathbf{E}(C)$, called the *probabilistic radius* of C, is the expected value of the evacuation radius of C over all the scenarios. Hence, we can define a new problem.

Probabilistic p-Center under Pressure (`PpCP`)

Instance:	A graph $G = (V, E)$, a set of scenarios S and a vector Π of probabilities associated to the scenarios of S, an integer p
Feasible solutions:	A p-set $C \subseteq V$ with $\mathbf{E}(C)$ finite.
Objective:	Find a solution of minimum probabilistic radius if it exists.

Note that, if an optimal solution exists, an optimal solution of size p always exists.

In this thesis, we focus on the analysis of PpCP with a uniform probability distribution over the scenarios. So, we will assume $\pi_s = \frac{1}{n}$ for all scenarios $s \in S$. In addition, we recall that we consider exclusively $S = V$. Therefore, to simplify our notations, we will refer to an instance (G, S, Π, p) of PpCP as (G, p). In this thesis, we may refer to both PpCP and RpCP as the UP problems.

2.1.5 Feasibility conditions

In this subsection we analyze necessary and sufficient conditions for a solution to be feasible on an instance (G, p) for both RpCP and PpCP. For both PpCP and RpCP, a solution C is said feasible if its objective value is finite, that is if the evacuation radius of C is finite for all scenarios. In the following, we consider without loss of generality that G is a connected graph. In addition, as the set of feasible solutions is similar for both problem, in this subsection we will refer to the set of feasible solutions for RpCP. We present first some additional notations and observations.

A *minimal articulation component*, or MAC for short, is an articulation component that does not strictly contain another articulation component. We denote $\Upsilon(G)$ the set of minimal articulation components. Note that an articulation component that is a singleton $\{v\}$ is necessarily minimal and this occurs if and only if v is a pending vertex. In addition, every vertex $a \in \mathcal{A}(G)$ is associated to at least 2 articulation components, and every articulation component is associated to one articulation point.

Lemma 1. *A is a minimal articulation component of G if and only if A is an articulation component which does not include an articulation point of G.*

Proof. $\Rightarrow$ By contrapositive we prove that if an articulation component A includes an articulation point, then A is not minimal. Let A be an articulation component induced by the articulation point $a \in V$. Suppose $b \in A$ is an articulation point of G. Then b induces at least two disjoint connected components in $G - b$. Since $b \neq a$, a is in one connected component of $G - b$, consequently $G \setminus A$ is a subset of this connected component. It follows that at least another component of $G - b$ is contained in A, which means that A is not minimal.

$\Leftarrow$ The proof is also by contrapositive. We prove that if A is a non-minimal articulation component, then A includes an articulation point. Let A an articulation component that is not minimal. Then there is an articulation component $B \subset A$ induced by the articulation point $b \in V$, such that $B \neq A$. Consider $x \in A \setminus B$ and $y \in B$. Since A is connected, x and y are connected in A by a path; this path necessarily crosses b and in particular $b \in A$. □

Lemma 2. *All minimal articulation components of G are pairwise disjoints.*

Proof. By contrapositive, we assume $A \in \Upsilon(G)$ and B an articulation component such that $B \neq A$ and $B \cap A \neq \emptyset$. We prove then that B is not minimal.

Let $x \in A \cap B$. Since A is a MAC, $B \not\subset A$. Then there is a vertex $y \in B \cap (V \setminus A)$. Every path between x and y in G crosses a. As B is a connected component, there is a path from x to y in B, thus $a \in B$, and B is not a MAC by Lemma 1. □

Given a graph $G = (V, E)$ and p, we denote with $\mathcal{C}_p(G)$ the set of feasible solutions of the RpCP-instance (G, p).

Proposition 3. Let (G, p) be an instance of RpCP with $|V| \geq 2$. A solution $C \subset V, |C| \leq p$ is in $\mathcal{C}_p(G)$ if and only if $|C| \geq 2$ and C includes at least one vertex in each minimal articulation component of G.

Proof. Suppose C is a feasible solution for RpCP on G. We have seen that C is a feasible solution for RpCP if and only if $r^s(C, j) \in \mathbb{R}, \forall j, s \in V$, i.e. all the evacuation distances over all vertices and all scenarios are finite.

First suppose there is no articulation point, then G has no articulation components. Let $s \in C$, and $x \in V, x \neq s$. In scenario s, x is assigned to a center that is not s. Thus $|C| \geq 2$. Conversely, if $|C| \geq 2$, for any scenario s, $G - s$ is connected and contains at least one center.

Second, suppose G has at least one articulation point and consequently at least 2 disjoint articulation components. In addition, if A is an articulation component of G induced by the articulation point a, then $\forall j \in A, r^a(C, j) \in \mathbb{R}$ if and only if $C \cap A \neq \emptyset$. Then C intersects all articulation components. In particular $|C| \geq 2$ and C intersects all minimal ones. Conversely, if C intersects all MACs then $|C| \geq 2$ and it intersects all articulation components since any articulation component contains a MAC.

□

Remark 1. For all UP problems considered in this thesis, feasibility conditions do not depend on the lengths, and the objective value is linear with respect to the lengths. As a consequence, the uniform case is equivalent to the case where all edge lengths are equal to 1.

Corollary 4. If G has at least 2 vertices, $\mathcal{C}_1(G) = \emptyset$.

As a consequence, from now we will consider only RpCP instances satisfying $p \geq 2$.

Corollary 5. For a given p, we can verify in polynomial time whether $\mathcal{C}_p(G) \neq \emptyset$.

Proof. For $G = (V, E)$, we generate $\mathcal{A}(G)$ in $O(|V| + |E|)$ using Tarjan's Algorithm ([32]). The minimal connected components of G are the connected components

of $G \setminus \mathcal{A}(G)$ adjacent to at most one articulation point in G, where a set V' of vertices is said adjacent to a vertex if this vertex has at least one neighbor in V'.

There is a feasible solution for `RpCP` on G if p is greater or equal to the number of MACs. □

Corollary 6. For all $C \in \mathcal{C}_p(G)$, C necessarily includes all pending vertices.

Proof. Every pending vertex is a MAC of G. Then by Proposition 3, a feasible solution includes all pending vertices. □

2.1.6 Examples and comparison with *p-Center*

We saw that a set $C \subset V$ is feasible for `RpCP` and `PpCP` if $r^s(C)$ is finite for all scenarios $s \in V$. $r^s(C)$ is finite if every vertex has a finite evacuation distance in scenario s. In some cases, most of the feasible solutions for `Min p-Center` are not feasible for the UP problems. For example, consider the path G given in Fig 2.5. For $p = 2$, any set of 2 vertices is a feasible solution for `Min p-Center` while the only feasible solution for the robust 2-Center problem is the set $\{1, 5\}$ of the extremity vertices of the path as proved previously in Section 2.1.5.. For any C without vertex 1, we necessarily have $r^2(C, 1) = +\infty$ since vertex 1 cannot reach any shelter. C is thus not feasible.

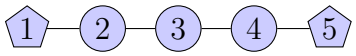


Figure 2.5: A path with 5 vertices and shelters on the extremities

In addition, when the optimal solution of `Min p-Center` is feasible for `RpCP` and `PpCP`, the relative error of using this solution for any of our problems can be arbitrarily large, as shown in the following. Consider the graph represented in Figure 2.6, with $M \geq 1$. Let $p = 2$. The optimal choice for `Min p-Center` is to locate centers at vertices 2 and 3, for a radius of 1. Consider now this solution in the context of `RpCP` and `PpCP`. The evacuation radius of $\{2, 3\}$ is $3, M+3, M+3, 3, M+1$ and $M + 1$ for each scenario $s = 1, 2, \ldots, 6$ respectively. Thus the robust radius of solution $\{2, 3\}$ is $M + 3$, and its probabilistic radius is $(4M + 14)/6$. However, if we choose instead to locate the shelters at vertices 5 and 6, the worst evacuation radius is 3 induced by scenarios 2, 3, 5 and 6. For scenarios 1 and 4, the evacuation radius is 2. Thus the robust radius of solution $\{5, 6\}$ is 3, and its probabilistic radius is $16/6$. Overall, in this example the relative error of using an optimal solution of `Min p-Center` for `RpCP` or `PpCP` is of order M.

Obviously, on an instance $(G, p), p \geq 2$ of `RpCP` or `PpCP`, adding a center to C cannot increase the evacuation radius. Therefore, it is straightforward that,

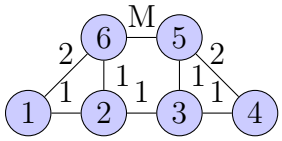


Figure 2.6: An example of different optimal solutions for `Min p-Center` and `RpCP` with $p = 2$.

if $|\Upsilon(G)| \leq p \leq |C|$, then there is an optimal solution with exactly p shelters. However, this is not a necessary condition for optimal solutions.

Consider indeed the graph of Figure 2.7 in the uniform case; if $p = 5$, then the minimum robust radius is 1, which is satisfied by $\{1, 3, 5, 7\}$. Similarly, $\{1, 3, 5, 7\}$ ensures a minimum probabilistic radius of value 1 for `PpCP` with $p = 5$. Note this is also the case with `Min p-Center`: if $p = 2$, then the minimum possible radius is 2, but $r(\{3\}) = 2$.

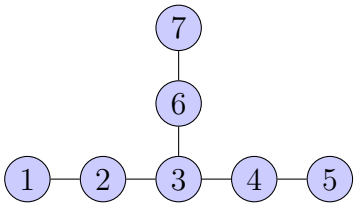


Figure 2.7: An example where, for $p = 5$, there is an optimal solution for `RpCP` and `PpCP` of size 4.

2.2 State of the Art

Location problems have long been receiving the attention of scientists. In 1909, Alfred Weber (younger brother of the well-known sociologist Max Weber) developed the first general theory of industrial location. His model took into account several spatial factors for finding the optimal location and minimal cost for manufacturing plants [33],[34]. In 1964, the network version of the problem, in which demand areas and facilities are located on an underlying graph representing the real world transportation system, becomes popular with the seminal work of Hakimi [35]. This work is often seen as the first step toward the development of the very active research field. Many surveys propose to categorize Facility Location problems according different key features, like the location space (continuous, network or discrete), the context (for example emergency logistics [36] or evacuation planning [30]), the different fundamental and advanced models [37], the decision-making environment [38] and many others.

In this section, we list some relevant related works in the literature for the `Min p-Center` problem and Facility Location problems in relation to shelter location problems for emergency situations. Unless otherwise stated, the results are on undirected graphs.

2.2.1 The p-Center Problem

The `Min p-Center` problem is a minimax facility location problem that consists in locating p identical facilities, also called *centers*, on a graph to minimize the maximum distance between demand vertices and a closest facility. The main concern of this problem is to minimize the negative impact of the worst case service level. This sort of objective is meaningful for problems with a time sensitive service structure, where the human life is at stake. Many applications arises in emergency service locations, such as determining optimal locations of police stations, hospitals, fire stations, ambulances and shelters [35, 37].

In [35], Hakimi identifies more precisely two separate problems: the *Weighted Vertex 1-Center* problem and the *Weighted Absolute 1-Center* problem. In the *Vertex* case, a center must be located on a vertex of the graph. In an instance of the *Absolute* case, an edge (i, j) of a graph $G = (V, E)$ can be considered as a set of infinite set of points joining vertices i and j. We can then identify a point x on a edge (i, j) by its distance from the endpoints of (i, j). In the *Absolute* case, a center can then be located on any point of the graph. In the weighted context, the distance of a vertex to the facility is weighted by the weight of the demand vertex. The maximal weighted distance to all vertices is then called the weighted absolute radius in the absolute case, or the weighted radius in the vertex case. The unweighted case can be reduced to the weighted case with uniform demand on the

vertices. The *Weighted Absolute* (respectively *Vertex*) *1-Center* problem is then to find a point (resp. vertex) whose maximal weighted distance to all vertices is minimum. This optimal point (respectively vertex) is called the weighted absolute (resp. vertex) center. The vertex center of G is not necessarily an absolute center: consider a graph of two vertices of weight 1 connected by an edge with length 1. In this example the absolute radius is 0.5 whereas the radius is 1. In [39], Hakimi generalizes the *Weighted/Unweighted Absolute/Vertex 1-Center* problems to the *Weighted/Unweighted Absolute/Vertex p-Center* problems.

Note that, for $p \in \{1, \ldots, n\}$, an instance (G, p) of the *Weighted Absolute p-Center* problem can be reduced to an instance of the *Weighted Vertex p-Center* as follows: An intersection point is a point x on an edge (i, j) if there exists two distinct vertices $u, v \in V$ such that x is the unique point on (i, j) for which the weighted distance from u to x is equal to the weighted distance from v to x. Denote P the set of induced intersection points of G. An optimal solution for *Weighted Absolute p-Center* can be found in the set of vertices and intersection points [40] [15]. We can then transform G in a vertex-weighted graph $G' = (V \cup P, E')$, where E' is the set of edges induced by the insertion of vertices P, and where the weight of $v \in V$ is the same in G and G', and the weight of $v \in P$ is 0 in G'. The solution for the *Weighted Vertex p-Center* problem on G' is a solution for the the *Weighted Absolute p-Center* problem on G.

For the sake of brevity, we have opted not to discuss the large body of literature on these different versions. As our definition of the `Min p-Center` as given in Section 1.2 corresponds to the *Unweighted Vertex p-Center* and is the most relevant to our work, we present in the following the main polynomial algorithms, complexity and approximation results valid for the *Unweighted Vertex p-Center*. The following complexity results assume that the distance-matrix of the graph is known, which would otherwise take $O(nm)$ to be computed [41].

We do not include in the following variants the *Capacited p-Center* problem, which is a variant where facilities are assigned capacities. In this case, a facility can not satisfies more demands than its capacity. As we consider territories with typically low density habitat (Subsection 2.1.1), we can safely consider that shelters are large enough [11].

2.2.1.1 Polynomial Cases, Complexity and Approximation Results

The problem of finding the *1-Center* can be done in $O(n^2)$ once the matrix of shortest distances for the graph has been computed: after computing the weighted distance-matrix, we must find a maximum entry in each row and then find a minimum over these maxima [15]. On a tree, Kariv and Hakimi propose in [15] an algorithm to solve the *1-Center* problem in $O(n \log n)$ time. Handler [42] improves this results with an $O(n)$ algorithm.

For the `Min p-Center` problem, Minieka is the first to propose a solution method with a finite series of `Min Set Cover` [40]: in order to minimize the number of shelters to be located to cover all vertices within a given radius r, we can use the `Min Set Cover` where $\Omega = V$ and $Z_i = \{j \in V : d(i,j) \leq r\}, \forall i \in V$. This specific problem corresponds to `Min r-Dominating Set` with radius r. If the optimal value of `Min r-Dominating Set` with radius r is greater than p, then the optimal value of `Min p-Center` is greater than r; otherwise, the optimal value is less or equal to r. Then, let $D = D_1, D_2, \ldots, D_t$ be an ordering of the distinct distance values in the matrix of shortest distances for G. We can solve `Min r-Dominating Set` with radius D_i, where D_i is taken in D in the decreasing order, until the optimal value of the optimal value is greater than p. Since $|D|$ is in $O(n^2)$, the number of steps is finite. This approach has been improved since by using a binary search strategy on $|D|$ which reduces the maximum number of steps to $O(\log n)$. In addition, to reduce the search space, lower and upper bound on the optimal value of `Min p-Center` can be computed. The complexity of the resulting algorithm depends hence on the complexity of `Min r-Dominating Set` on the given class of graph. More precisely, if on a given graph `Min r-Dominating Set` is in $O(f(n))$, one can solve `Min p-Center` on that graph in $O(f(n) \log n)$.

In [15], for $p \geq 2$, Kariv and Hakimi proved the NP-hardness of `Min p-Center` on an arbitrary planar graph G of maximum degree 3 and all edge lengths equal to 1. The proof is based on the relation between `Min p-Center` and `Min Dominating Set`. We present a few key ideas of this relation, useful in our thesis. An optimal solution for the `Min p-Center` problem with $p = \gamma(G)$ is an optimal solution for the `Min Dominating Set`. However, the `Min Dominating Set` and the problem of finding the domination number are NP-hard [25]. In addition, `Min Dominating Set` is equivalent to `Min r-Dominating Set` with radius 1. Consequently, `Min r-Dominating Set` with radius 1 on a graph $G = (V, E)$ can be reduced to the `Min Dominating Set` on a graph (V, E^r), where, for vertices $i, j \in V$, $(i, j) \in E^r$ if and only if $d(i, j) \leq r$ in G.

The same year, Hsu and Nemhauser [43] proved it is NP-hard to find an approximation algorithm in a metric space with ratio less than 2. Since then, several 2-approximation algorithms were developed [44], [45], [46], [47]. Recently, Garcia-Diaz & all. proposed in [48] a survey and an analytical study and experimental evaluation of the most representative approximation algorithms for `Min p-Center`.

On a tree, Kariv and Hakimi proved in [15] that `Min p-Center` can be solved in $O(n \log^{p-1})$. If $p = 2$, Handler proposed an $O(n)$ algorithm to solve `Min p-Center` [49].

2.2.1.2 Exact solutions and heuristics

Apart from the `Min r-Dominating Set` approach as described previously, other exact solution methods have been proposed.

Daskin [50] proposed the first mixed integer programming (MIP) formulation for `Min p-Center`. Elloumi, Labbé and Pochet [51] proposed another IP formulation using the fact that the optimal value of `Min p-Center` is restricted to a finite set of distance values. Calik and Tansel [52] developed two new IP formulations, the second one being a tightened formulation using a relationship between their first formulation and the formulation proposed by Elloumi et al. [51]. We will describe these formulations and corresponding algorithms more extensively in Chapter 6.

Finally, we present some heuristics for `Min p-Center`. Mladenovic et al. [53] introduced the first meta-heuristic approaches for finding approximate solutions to `Min p-Center`. They proposed a multistart local search algorithm, a chain substitution Tabu Search algorithm, and a variable neighborhood search algorithm. While conducted experiments on the instances from the OR-Library among other, their algorithm outperformed Hochbaum and Shmoys' [44] algorithm. Pullan [54] proposed a genetic algorithm for the `Min p-Center` which combines a population based meta-heuristic with a local search algorithm. Salhi and Al-Khedhairi [55] obtained light lower and upper bounds by using three-level meta-heuristic and integrated these bounds with the algorithm by Daskin [50] to solve the `Min p-Center` relatively more quickly.

Martinich [56] proposed a method called the vertex-closing approach for the `Min p-Center` on complete networks with distance values satisfying the triangle inequality. Initially, the algorithm places a facility on each vertex. Then the problem is to find $(n - p)$ facilities to be closed, so that the maximum of the distances between the vertices and their facilities is minimized. They proposed two polynomial algorithms of complexity $O(m \log m)$ and $O(m^2)$. The algorithms are proven to converge to an optimum for special cases, and computational experiments suggest that they produce very good solutions. In particular both algorithms perform very well on problems where p is large relative to the number of vertices n, specifically when $p/n \geq 0.30$.

Mihelic and Robic [57] solved the `Min p-Center` on complete networks with distance values satisfying the triangle inequality by introducing a polynomial time heuristic algorithm based on solving a finite series of minimum dominating set. They experimentally showed that their algorithm performs much better than other existing heuristics (at that time) and is competitive with the best known (non-polynomial time) algorithms for solving `Min p-Center` in terms of average quality and deviation of the results as well as execution time.

Finally, Bozkaya and Tansel [58] showed that there exists a spanning tree of

any connected network such that the optimal solution to the `Min p-Center` on this tree is optimal also for the network under consideration. They experimented on two classes of spanning trees to observe how often these trees provide the optimal solution. They concluded that these two classes of spanning trees do not always include the optimizing tree, but they do in most of the instances.

2.2.1.3 Other Fundamental Facility Location problems

The fundamental problems of Facility Location include `p-Median` problems, *Fixed-Charge Facility Location* problems and *Covering Location* problems. The following definitions are based on the book by Laporte, Nickel and da Gama [37] and on the recent survey by Turkoglu and Genevois [59].

The `p-Median` problem
The `p-Median` problem is the problem of finding a location for p facilities that minimizes the sum of distances between a vertex and the nearest facility, over all vertices. Kariv and Hakimi proved in [15] that `p-Median` is NP-hard on a general graph. Often the objective function focuses rather on the costs, which are directly proportional to the distance and can be weighted with regard to the demand vertex.

The *Fixed-Charge Facility Location* problem
The *Fixed-Charge Facility Location* problem is a minisum problem like `p-Median`, where the number of facilities that can be opened is not constrained. However the opening costs of facility opening are added to the objective function. *Fixed-Charge Facility Location* aims at minimizing the total cost (opening and travel).

Covering Location problems
A covering location problem deals with situations where the service provided by a facility is ensured only within a predefined distance of it. For example, if the facilities correspond to ambulance stations, an area is considered covered only if the ambulance can arrive there in less than 5 minutes. There are two main types of covering location problems: the first one corresponds to `Min r-Dominating Set` as a particular case of `Min Set Cover`, as previously described. `Min Set Cover` has been proved NP-hard by Garey and Johnson in [25] and is seen as one the three special structures in pure integer programming with the most wide-spread applications. The second type of covering location problems is *Maximal Covering*, where the number of facilities that can be located is bounded. Therefore *Maximal*

Covering aims to maximize the covered vertices within the restricted number of facilities that can be located.

2.2.2 Models under indeterminacy

Facility location problems in nondeterministic environments are generally described in two stages: first, before the indeterminacy is resolved, we can choose locations. Then, once indeterminacy has been resolved, react for example by assigning vertices to facilities. This description matches real life situations like facility breakdown, natural disaster cutting off communication, etc., and it has already been studied under different setups. Therefore, facility location problems have been subject to many approaches for decision making with indeterminacy (see for example the reviews [38, 36, 37]). More specifically, several models for `Min p-Center` have already been developed for environments where indeterminacy is taken into account.

We propose in this section a brief presentation of some `Min p-Center` variants under indeterminacy, as well as some other models relevant for our context. When appropriate, we will outline the differences between the introduced work and the UP model. In the following, we consider two analysis axis: the origin of the indeterminacy and the decision-making environment.

Origin of the indeterminacy

When addressing a facility location problem under indeterminacy, indeterminacy can come from different parts of the input data [60] [61]:

- *provider-side* indeterminacy corresponds to indeterminacy in facility capacity, facility reliability, facility availability, etc.
- *receiver-side* indeterminacy is related to the uncertain structure of the set of users, users demands, users locations etc.
- *in-between* indeterminacy refers to the lack of complete knowledge about the transportation network topology, transportation times or costs between facilities and users.

Decision-making environment

Conventionally, one distinguish three decision-making environments: certainty, risk and uncertainty [16]. In certainty situations, all parameters are deterministic and known, and there is no element of chance between the decision and the

outcome. On the other hand, risk and uncertainty both involve randomness in the parameters. In risk situations, the value of some parameters are governed by probability distributions that are known by the decision maker. In uncertainty situations, no probabilistic information is used either because it is not available or because the decision maker prefers not to resort to it.

In Table 2.1, we classify the works we will present following this double analysis grid. We present first some relevant variants of `Min p-Center` under indeterminacy with regard to the origin of the indeterminacy. We think this approach is the most relevant to outline the difference between these models and the UP model. We will also discuss why our model fits in the *in-between* indeterminacy type. Secondly, we will categorize these works and discuss some other relevant works from the point of view of the decision-making environment. Finally, we will discuss the representation of indeterminacy with respect to our model.

Type / Environment	Certainty	Uncertainty	Risk
Provider-side	[62, 63, 64]	[60, 65]	
Receiver-side	[64]	[66, 67, 68, 69, 65]	[70]
In-between		`RpCP` [71, 72, 69, 65]	`PpCP`

Table 2.1: Review

2.2.2.1 About the origin of the indeterminacy

Provider-side indeterminacy

Provider-side indeterminacy may relate to randomness in the facility availability or reliability. For example, the `q-Neighbor Min p-Center` problem [62, 63], also called *Fault tolerant* `Min p-Center` problem [73], is a generalization of `Min p-Center` where, given a number q, we have to place p centers so as to minimize the maximum distance of any non-center vertex to its q^{th} closest center. More precisely, each vertex is assigned q-service facilities, so that each demand vertex could withstand the failure of $q-1$ service facilities. Chaudhuri, Garg and Ravi give in [62] a 2-approximation algorithm for this problem, and show it is the best possible. Other exact algorithms and experimental results are proposed in [63] by Chen and Chen. In the UP model, in some scenarios the fire occurs in an area with a shelter. We have seen that in this case the shelter is no more reachable for users outside the area (user inside the disaster area can get into the shelter). This situation corresponds to a provide-side type of indeterminacy. However, note that the evacuation strategy, even in these situations, does not correspond to an automatic reassignment to a q^{th} center, for some predetermined q. In the UP model,

we do not assign vertex v to some predetermined q^{th} closest center in G, because, among other things, we have no guarantee that the q^{th} closest center for v in G is accessible in G^s. If we consider a graph G^s, while the vertices $v \in V \setminus \{s\}$ are evacuated to their closest center in G^s, the evacuation strategy of vertex s depends on its neighborhood.

Receiver-side indeterminacy

Receiver-side indeterminacy may express randomness in the demand. For example, Bhattacharya, Kameda and Song [66] consider the *minimax regret* [1] `Min 1-Center` with uncertainty restricted to vertex weights. They introduce an $O(n)$ algorithm for path, an $O(n \log n)$ algorithm for trees, an $O(n \log n)$ algorithm for cycle and an $O(n \log^2 n)$ algorithm for cactus. Martínez-Merino et al. [70] consider a `Min p-Center` variant with probability on vertex demand on an Euclidean plane. Averbakh and Berman [67] propose an exact algorithm for *minimax regret* `Min 1-Center`. It leads to a $O(n^2 \log^2 n \log \log n)$ algorithm for trees, and a $O(mn^2 \log n)$ algorithm for general graphs. In [68], Taghavi and Shavandi consider the `Min p-Center` problem with interval associated to vertex weights.

In-between indeterminacy

In-between indeterminacy corresponds to randomness in travel time or transportation cost for example. Many models consider indeterminacy on the weight (seen as cost or travel time) associated to the edges. The objective function focuses then on the weighted distance. In [71] by Averbakh and in [72] by Lu, a robust version of `Min p-Center` is defined by introducing uncertainty on edge weights: each weight may vary in an uncertainty set (usually an interval) and the problem is to determine a solution minimizing the worst case or the maximum regret. In [71], Averbakh proved that the corresponding minimax regret `Min 1-Center` with uncertain edge weight is strongly NP-hard, in contrast with the problem with uncertain vertex-demand weight.

Note that, a model can incorporate different types of indeterminacy. For example, in [69], Averbakh and Berman consider the *minimax regret weighted* `Min 1-Center` on a network with uncertainty in node weights and edge lengths. They showed this problem can be solved on trees with a $O(n^2 \log n)$ algorithm.

In [65], the authors Du, Zhou and Leus propose a robust minimax model for a

[1] Under a minimax regret objective function, the optimal decision is that for which the largest difference of performance (called the regret) for each scenario between a solution and the best solution for the scenario is minimized. More details in subsection 2.2.2.2.

reliable `Min p-Center` problem. Each scenario corresponds to a set of disrupted facilities and updated demands and costs. Clients are reallocated to the nearest surviving facility. This model incorporates the three types of indeterminacy. We will come back to their model later on to outline its difference with the UP model.

Huang, Kim and Menezes propose in [64] a variant of the `Min p-Center` problem for large-scale emergencies, where the disaster affects a single vertex s, including any facility on this vertex. This model incorporates both indeterminacies in the facility availability and in the demand: any facility on an affected vertex is no longer available but only the population on this vertex requires evacuation. This model is motivated by different kind of disasters that affect a single vertex but also by the fact that each vertex corresponds to a large zone, like an entire city. Our context is really different since all zones must be evacuated in each scenario s and a shelter always secures at least the people from the corresponding area.

The UP model corresponds to an in-between type of indeterminacy. One of the main differences between the UP model and all known models is that we apply indeterminacy on the graph structure and not on the values of the edges between the vertices. We think that considering rather indeterminacy over the vertex structure of the graph will allow us to better apprehend the impact of the fire on the transport network. In some of these works, nondeterminate parameters vary independently one from the other. In our context, this independence hypothesis is not relevant since, if a fire ignites on a vertex, then all weights of the edges incident to this vertex are modified in the same way.

2.2.2.2 About the indeterminacy model

Certainty situations

In certainty situations all parameters are deterministic and known, and there is no element of chance between the decision and the outcome. We apply deterministic optimization to find an optimal decision with the use of an objective or multi-objective function. This environment is not necessarily unsuitable for managing situations which in practice include indeterminacy. In this case, the deterministic models generally adopt a single hazard scenario such as a worst-case or a most probable scenario. A vast literature in evacuation planning mostly relies on such deterministic models (see for example the review in [30]), as well as in emergency logistics (see for example the review in [36]). The `Min p-Center` problem for large-scale emergencies[64] and the the q-Neighbor `Min p-Center` problem [62],[63] previously introduced correspond also to this case.

Risk situations
In risk situations probability information on the parameters is available. In this case, stochastic optimization is usually used, generally to optimize the expected value of a given objective function, or to maximize the probability that the solution is "good". The problem is then often solved either by using a specially designed algorithm (for example [70]), or using the general *Stochastic Programming* (SP) techniques. In a SP problem, there are first stage decision variables and second stage decisions variables. First stage decision variables must be set before risk has been resolved, while second stage decisions variables have to be set after the risk has been resolved. Among SP problems for facility locations, Bayram and Yaman propose for example a stochastic evacuation planning model in [74] that optimally locates shelters and assigns evacuees to the nearest shelter so as to minimize the expected total evacuation time. In [60], the authors propose a SP problem to solve a variant of *Fixed-Charge Facility Location* with probability on the accessibility to the facility. Note that `PpCP` falls into the paradigm of stochastic optimization but not directly of SP. In `PpCP`, first stage decision variables correspond to the shelter location; while second stage decision variables correspond to the evacuation path followed by each person on the territory. Unlike the shelter location, the choice of the evacuation path followed by a person is not under control of the decision-maker.

Uncertainty situations
In uncertainty situations no probability information on the parameters is used. In this case, a robustness measure is usually considered for evaluating the performance of the system. There are two classical objective functions: minimax and minimax regret [16, 38, 75]. In a minimax problem, an optimal solution minimizes the maximum value of the solution across all scenarios (for example [65]). Minimax regret problems are based on the definition of the regret of solution. There are two common measures: the first one considers the difference between the value of the chosen solution in a given scenario and the value of the optimal value for that scenario (all of the robust problems based on minimax regret cited in Subsection 2.2.2 refer to this definition). The second one measures the ratio between the two previously mentioned values. Note that `RpCP` falls into the paradigm of minimax robust optimization. In addition, our model includes a layer of minimax robust optimization regarding the measured evacuation distance of the fire-source area: in scenario s, if no shelter is located on s, then the evacuation distance is given by the longest evacuation path originating from s.

2.2.2.3 About the indeterminacy representation

In some models, non determinate parameters may vary independently one from each other (for example in [67, 68, 70]). In our context, this independence hypothesis is not relevant since, if a fire ignites on a vertex, all length of the edges incident to this vertex are modified in the same way. In the UP model, indeterminacy is represented by a set of discrete scenarios. Such scenario-based approach has already been studied for facility location problems [38, 75]. Most deal with scenario-based approaches for generalizations of `p-Median` and *Fixed-Charge Facility Location* problems. For example, in [76], a `p-Median` problem under scenario-based demand uncertainty is considered. In [77], a new robustness measure is introduced and used on a discrete scenario-based approach for the `p-Median` problem. In [60], the authors consider a scenario-based variant of *Fixed-Charge Facility Location*, where each scenario specifies the set of second-stage accessible facilities. The two reviews [38, 75] underline that robust `Min p-Center` problems are often more difficult than the related `p-Median` problems. To our knowledge, the only paper studying the robust `Min p-Center` problem with a scenario approach is the previously presented model by Du, Zhou and Leus [65]. `RpCP` differs from this model, as it can not be reduced to a robust minimax `Min p-Center` with a scenario-based approach, where each scenario s defines the length of the distance between two vertices in G^s: the evacuation distance $r^s(C, v)$ depends on the solution and does not systematically correspond to a shortest path in G^s.

Conclusion

In this chapter we define the problems studied in this thesis. Our objective is to determine a best location for shelters in a territory threatened by wildfires. The territory, divided into zones, is represented by a graph in which each zone corresponds to a vertex and two vertices are linked by an edge if it is possible to go directly from one zone to the other. The problem is to locate at most p shelters on vertices so that the maximum evacuation distance of any vertex to its nearest shelter is minimized. When the uncertainty of fire outbreaks is not considered, this problem corresponds to the well-known `Min p-Center` problem on a graph. A scenario defines a fire outbreak on a single zone with the main consequence of modifying evacuation paths. Several evacuation paths may become impracticable and the ensuing evacuation decisions made under pressure may no longer be rational. We introduce a new evacuation strategy, the UP evacuation strategy, to characterize this behavior. This evacuation strategy induces new evacuation distances which render our approach specific. The evacuation distance of this source area is then the length of a longest evacuation path leaving this area. The evacua-

tion distance of any other area is given by a shortest operational path to a shelter. The evacuation radius of a p-set for a scenario is then the maximum evacuation distance in this scenario. We define two problems; they vary by the input data and the objective function considered. In `PpCP`, we are in a risk environment and the objective function is to minimize the probabilist evacuation radius. In `RpCP`, we are in an uncertain environment and the objective function is to minimize the robust evacuation radius. A main objective in our context is to avoid any death toll, thus a p-set is considered feasible only if the evacuation distances of all vertices in all scenarios are finite. This constraint is verified for both `PpCP` and `RpCP` when $p \geq 2$ and the p-set intersects each minimal articulation components of the graph. A direct result is that a shelter must be located on each pending vertex.

Compared to the existing state of the art, our problems fall within the scope of two-stage problems with in-between indeterminacy. To our knowledge, it is the first attempt to model the evacuation strategy in such manner for a shelter location problem.

In the next chapter, we study the NP-hardness of our two new problems.

CHAPTER 3

HARDNESS RESULTS

Abstract

In this chapter, we study the hardness of the problems RpCP and PpCP. We show that RpCP is NP-hard in all hereditary classes of graphs where the decision version of Min Vertex Cover is NP-complete, and NP-hard in subgrids of degree at most 3. In addition, we give a non approximation result for PpCP on bipartite planar graphs and another non approximation ratio on subgrids of degree at most 3.

Contents

3.1 RpCP Hardness results 43

3.1.1 $\texttt{RpCP}_1$ and the Min Vertex Cover problem 44

3.1.2 $\texttt{RpCP}_2$ and the Min Dominating Set problem 46

3.1.3 $\texttt{RpCP}_2$ and the Min Vertex Cover problem 47

3.1.4 Does increasing the radius make the decision problem harder? 52

3.2 PpCP Non-Approximation Results 54

3.2.1 PpCP Non-Approximable on Planar Graphs 54

3.2.2 PpCP Inapproximation subgrids 57

Introduction

In the previous chapter, we introduced two new problems for shelter location: PpCP and RpCP. In this chapter, we analyze the hardness of these two problems. Note that if p was a fixed constant, then the number of subsets of size p would be polynomial and these problems could be polynomially solved on any graph (even thought not tractable in practice). We saw that Min p-Center is NP-hard on planar graphs of maximum degree at least 3. This result does not imply immediately the hardness of PpCP and RpCP. Indeed, we defined our model with non-zero probabilities, which does not count the classic Min p-Center as one of its specific cases.

For a graph $G = (V, E)$, denote $L \subset V$ the set of pending vertices. Given the feasibility conditions seen in Subsection 2.1.5, we assume $p \geq |L|$ and L contained in any feasible solution for PpCP and RpCP. Thus, when $p = |L|$, we can straightforwardly say that L is the only feasible solution for both PpCP and RpCP. Therefore, we consider in the following that $p > |L|$.

The case of planar graphs of low degree is particularly relevant for real case applications. Since the graph G represents the adjacency graph of zones in the territory, it is planar and in most cases, each zone has a small number of adjacent zones. In some cases however, the underlying graph has even a simpler structure. A common case is a rectangular grid or subgrid when the territory has some "holes" corresponding to large spaces difficult to traverse, like a lake, steep slopes or marshland. Based on these cases, a natural question is the complexity of RpCP and PpCP in bipartite planar graphs (since subgrids are bipartite) and subgrids. In what follows, we answer these questions.

In Section 3.1 we present some hardness results for RpCP. A significant portion of the work presented in this section has been published in [28]. In Section 3.2 we present two hardness-in-approximation results for PpCP, one on bipartite planar graphs with degrees 2 or 3, and one on subgrids with degrees 2 or 3. The work presented in this section has been published in [27] (result on bipartite planar graphs) and have been submitted [78] (result on subgrids).

In this chapter, we present different transformations of graphs. Given an initial graph $G = (V, E)$, our transformations correspond always to an embedding of G into a graph $G' = (V', E')$. For clarity, for every vertex u of G, we will denote as well by u the vertex of G' it maps to. Using this convention, we consider that the vertex set of G is a subset of the vertex set of G'. A specific transformation considered in this chapter is a grid embedding: given a subgrid $G = (V, E)$, a grid embedding is a one-to-one function from V to V' for some dimensions (n, m) such that every edge $(u, v) \in E$ maps to an edge of the $n \times m$ grid. Unless otherwise stated, each time we will refer to a subgrid, we will assume that a grid embedding is given. As defined in [79], for a partial subgrid G and a positive integer f, the f-

expansion of G, denoted $Exp(G, f)$, is obtained from G by inserting $f-1$ vertices on each edge (each edge becomes a path of f edges). If $f \geq 2$, the *f-expansion* of any partial subgrid is a subgrid. If G is a subgrid embedded in a $n \times m$ grid $\mathcal{G}$, then $Exp(G, f)$ is a subgrid embedded in the $[(n-1)f+1] \times [(m-1)f+1]$ grid $Exp(\mathcal{G}, f)$. The vertex set of G can be seen as a subset of the vertex set of $Exp(G, f)$ and more precisely, in the related grid embedding of $Exp(G, f)$, the coordinates of any vertex $u \in V$ are multiplied by f compared to its coordinates in the original grid embedding of G in $\mathcal{G}$.

The following Lemma is certainly a known remark but we show it since we did not find any reference for it.

Lemma 7. The `Min Vertex Cover` problem is NP-hard in planar graphs with vertices of degree 2 or 3.

Proof. The decision version of the `Min Vertex Cover` is known to be NP-complete on planar graphs of maximum degree 3 [15]. Consider a planar graph G of maximum degree 3 with a pending vertex v adjacent to vertex y. Consider the graph G' obtained from G by adding a triangle $\{v_1, v_2, v_3\}$ and linking v_1 with v (v is then of degree 2 in G'). G' is planar with maximum degree 3 and with one pending vertex less than G.

We prove that G has a vertex cover of size t if and only if G' has a vertex cover of size $t+2$, which concludes the proof.

$\Rightarrow$ Let U be a vertex cover of size t on G. We build a vertex cover U' for G' based on U by adding v_1, v_2 which cover the edges of $G'[\{v, v_1, v_2, v_3\}]$. U' is then a vertex cover for G' of size $t+2$.

$\Leftarrow$ Let U' be a vertex cover of size $t+2$ on G'. Notice that U' includes at least 3 vertices of $\{y, v, v_1, v_2, v_3\}$. We can always chose to include y, v_1, v_2 in U' instead of any three vertices of $\{y, v, v_1, v_2, v_3\}$ without increasing the size of U'. As $y \in U'$ covers the edge (v, y), $U' \setminus \{v_1, v_2\}$ is a vertex cover of $G' \setminus \{v_1, v_2, v_3\}$. Then $U = U' \setminus \{v_1, v_2\}$ is a vertex cover for G of size t. □

3.1 RpCP Hardness results

For this section, we consider the version of `RpCP` with an evacuation radius threshold, `RpCP`$_k$. For any constant integer k, `RpCP`$_k$ takes as input an instance of `RpCP`. This problem is to decide whether there is a solution C with a robust radius less or equal to k, i.e. $\mathbf{R}(C) \leq k$. This problem is clearly in NP: we consider a polynomial number of scenarios and consequently, for any solution C, checking whether $\mathbf{R}(C) \leq k$ can be done in polynomial time. For each scenario, it requires evaluating the evacuation distances of each vertex using a minimum path algorithm. Note

that the set of problems $(\texttt{RpCP}_\texttt{k})_{k\geq 1}$ for any integer k corresponds to the decision problem of RpCP.

In this section, we will address $k = 1$ or 2 and the uniform case for different classes of graphs. In Section 3.1.1, we outline the relation between $\texttt{RpCP}_1$ and the decision version of the `Min Vertex Cover` problem. In Section 3.1.2, we outline the relation between $\texttt{RpCP}_2$ and the decision version of the `Min Dominating Set` problem. In both cases, we can deduce hardness results for $\texttt{RpCP}_1$, $\texttt{RpCP}_2$ and RpCP on some classes of graphs. In Section 3.1.3, we consider a particular subclass of bipartite planar graphs, the class of subgrids of maximum degree 3. $\texttt{RpCP}_1$ is polynomially solvable in this class while $\texttt{RpCP}_2$ reveals to be NP-complete. We conclude in Section 3.1.4 with some comments about the complexity of $\texttt{RpCP}_k$ when k varies.

3.1.1 $\texttt{RpCP}_1$ and the `Min Vertex Cover` problem

We consider an undirected graph $G = (V, E)$ where all edges are of length 1. The following elementary proposition leads immediately to a first hardness result:

Proposition 8. The robust radius of a set of vertices $C \subseteq V$ is at most 1 if and only if C is a vertex cover that includes all pending vertices.

Proof. Suppose C is a set of vertices with $\mathbf{R}(C) \leq 1$. Since its robust radius is finite, it should include all pending vertices. Let $u \in V \setminus C$, then $r^u(C, u) = 1$, which means that all neighbors of u are in C. Thus, C is a vertex cover.

Conversely, assume C is a vertex cover including all pending vertices; we consider a scenario u and a vertex v such that $v \notin C$ (in particular v is not a pending vertex). If $v = u$, then every neighbor of v is in C since $v \notin C$. If $v \neq u$, then v has at least one neighbor in $C \setminus \{u\}$ since v is of degree at least 2. In both cases, the evacuation distance is 1 and consequently $r^u(C) = 1$. Therefore, $\mathbf{R}(C) \leq 1$, which completes the proof. □

In particular, if G has no pending vertex, then $\mathbf{R}(C)$ is at most 1 if and only if C is a vertex cover. We deduce immediately:

Corollary 9.

1. $\texttt{RpCP}_1$ is NP-complete in all classes of graphs in the uniform case and of minimum degree 2, for which the decision version of `Min Vertex Cover` problem is NP-complete. The problem RpCP is NP-hard in these classes of graphs.

2. $\texttt{RpCP}_1$ is polynomial-time solvable in all classes of graphs in the uniform case and of minimum degree 2, for which the decision version of `Min Vertex Cover` problem is polynomial-time solvable.

For hereditary[1] classes of graphs $\mathbb{H}$, we can easily use a pre-processing allowing to reduce the `Min Vertex Cover` problem in this class to the same problem in the subclass of graphs in $\mathbb{H}$ without pending vertices. This leads to the following corollary.

Corollary 10.

1. $\texttt{RpCP}_1$ is NP-complete in all hereditary classes of graphs for which the decision version of `Min Vertex Cover` problem is NP-complete. The problem `RpCP` is NP-hard in these classes of graphs.

2. $\texttt{RpCP}_1$ is polynomial-time solvable in all hereditary classes of graphs for which the decision version of `Min Vertex Cover` problem is polynomial-time solvable.

Proof. (1): Suppose $\mathcal{H}$ is an hereditary class of graphs for which the decision version of `Min Vertex Cover` is NP-complete. We show `Min Vertex Cover` reduces to $\texttt{RpCP}_1$ in this class. Consider (G, p), with $G \in \mathcal{H}$ and $p \in \mathbb{N}$ an instance of the decision version of `Min Vertex Cover`. We transform G using the following pre-processing: while the graph has a pending vertex x and $p \geq 0$: add to the cover the unique neighbor y of x, remove from the graph (x, y) and all edges incident to y and subtract 1 to p. If p becomes negative, then the original instance was a no-instance If all vertices of G are removed and $p \geq 0$, then it was a yes-instance. In other cases, if (G', p') are the resulting graph and constant, then the original instance is a yes-instance if and only if (G', p') is a yes-instance. Since G' is in $\mathcal{H}$ and has no pending vertex, this is equivalent to deciding whether there is in G' a p'-center of robust radius at most 1.

(2): Suppose $\mathcal{H}$ is an hereditary class of graphs for which the decision version of `Min Vertex Cover` can be solved in polynomial-time and consider an instance (G, p), with $G \in \mathcal{H}$ and $p \in \mathbb{N}$ of $\texttt{RpCP}_1$. Using Proposition 8, it is a yes-instance if and only if there is a vertex cover including all pending vertices and of cardinality at most p. If G has p' pending vertices, with $p' > p$, then it is a no-instance. If $p' \leq p$ now, then this is equivalent to say that the subgraph obtained by removing the pending vertices and the related incident edges has a vertex cover of size $p - p'$. This question can be answered in polynomial-time since the resulting graph is in $\mathcal{H}$. □

In particular, deciding whether the minimum robust radius is 1 ($\texttt{RpCP}_1$) is NP-complete on planar graphs of maximum degree 3 since the decision version of `Min Vertex Cover` is NP-complete on this hereditary class [80].

[1]We call *hereditary* a class of graphs such that, for any graph in the class, all its subgraphs are still in the class.

3.1.2 $\texttt{RpCP}_2$ and the `Min Dominating Set` problem

We consider an undirected graph $G = (V, E)$ where all edges are of length 1. The following proposition leads to a first hardness result for $\texttt{RpCP}_2$:

Proposition 11. Let $G = (V, E)$ be a triangle-free graph. D is a dominating set of G that includes all pending vertices of G if and only if

$$\mathbf{R}(D) \leq 2 \Leftrightarrow r^s(D, v) \leq 2, \forall v \in V, \forall s \in V$$

Proof. Suppose D is a dominating set including all pending vertices, we consider a scenario s and a vertex v such that $v \notin D$ (in particular v is not a pending vertex). If $v = s$, consider then the evacuation path of v crossing any neighbor y of v. Vertex y is either in D or adjacent to a vertex of D which cannot be s. In this case $r^s(D, v) \leq 2$. If v and s are not adjacent, then v has a neighbor in D and $r^s(D, v) = 1$. The same holds if v and s are adjacent with $s \notin D$. Finally, if v and s are adjacent with $s \in D$, then there is at least one neighbor $y \neq s$ of v who is at a distance at most 1 of a vertex $x \in D$, where $x \neq s$ since G is triangle-free. In this case $r^s(D, v) \leq 2$. Consequently, $r^s(D, v) \leq 2, \forall v \in V, \forall s \in V$.

Conversely, suppose D is a set of vertices with $r^s(D, v) \leq 2, \forall v \in V$ and $\forall s \in V$, which is equivalent to statement $\mathbf{R}(D) \leq 2$. Since every evacuation distance is finite, D includes all pending vertices. Take a vertex $v \notin D$ and $u \notin D$ an vertex adjacent to v. Then, for the scenario u, an evacuation path through v leads to a vertex shelter adjacent to v. Therefore, every vertex of G is either in D or adjacent to D. Consequently, D is a dominating set that includes all pending vertices. □

In particular, if G is triangle-free of degree at least 2, then $\mathbf{R}(C)$ is at most 2 if and only if C is a dominating set. We deduce:

Corollary 12.

1. $\texttt{RpCP}_2$ is NP-complete in all triangle-free classes of graphs in the uniform case and of minimum degree 2, for which the decision version of `Min Dominating Set` problem is NP-complete. The problem `RpCP` is NP-hard in these classes of graphs.

2. $\texttt{RpCP}_2$ is polynomial-time solvable in all triangle-free classes of graphs in the uniform case and of minimum degree 2, for which the decision version of `Min Dominating Set` problem is polynomial-time solvable.

3.1.3 $\texttt{RpCP}_2$ and the `Min Vertex Cover` problem

In this section, we present a reduction from `Min Vertex Cover` to $\texttt{RpCP}_2$ which proves the NP-completeness of $\texttt{RpCP}_2$ in subgrids with vertices of degree 2 or 3. The purpose of this section is mainly to introduce the structural relations between these two problems, which will be needed in the next section when proving the NP-hardness of `PpCP` in subgrids of maximum degree 3.

To make the presentation easier to read and the proof more intuitive, we first propose a weaker result (Proposition 13) stating the hardness of $\texttt{RpCP}_2$ in bipartite planar graphs of maximum degree 3. Then, we show how to extend the reduction with a more complex construction to prove Theorem 15 that states the hardness of $\texttt{RpCP}_2$ in subgrids with vertices of degree 2 or 3. This step uses ideas and techniques proposed in [81, 79] to prove NP-hardness results in subgrids for a large range of problems known to be hard in planar bipartite graphs with vertices of degree 2 or 3.

Proposition 13. $\texttt{RpCP}_2$ is NP-complete in planar bipartite graphs with vertices of degree 2 or 3, even in the uniform case. `RpCP` is NP-hard in this case.

Proof. We revisit a reduction from the decision version of `Min Vertex Cover` to `Min Dominating Set` in planar graphs with vertices of degree 2 or 3, shown to be NP-complete in [25]. Given a planar graph $G = (V, E)$ with degrees 2 or 3, one builds a bipartite graph $G' = (V', E')$ by replacing each edge (u, v) with a gadget L_4 as presented in Figure 3.1. G' has vertices of degrees 2 and 3.

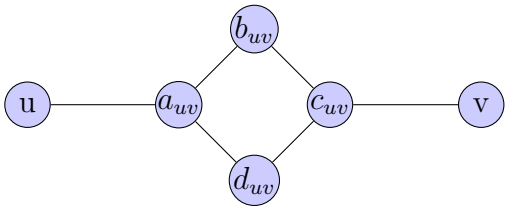


Figure 3.1: The gadget L_4 for an edge (u, v)

Lemma 14. For any $t \leq |V|$, G has a vertex cover of size t if and only if G' has a $(t + |E|)$-set C with $\mathbf{R}(C) \leq 2$. Moreover, for each edge $(u, v) \in E$, C has exactly one vertex in $\{a_{uv}, c_{uv}\}$ and none in $\{b_{uv}, d_{uv}\}$.

Proof. $\Rightarrow$ Suppose that G has a vertex cover $U \subset V$ of size t. We add to it a set $U' \subset V'$ of vertices to make it the required solution. To make the following construction non-ambiguous, we consider an orientation of the graph G. Consider any edge (u, v) oriented from u to v. As U is a vertex cover, at least one of the

endpoints of (u, v) is in U. If $u \in U$, then we add c_{uv} to U'. Otherwise, we add a_{uv} to U'. As a result, $C = U \cup U'$ is a $(t + |E|)$-set and as there is no pending vertex in G', every vertex has an evacuation distance of value at most 2. Thus, $\mathbf{R}(C) \leq 2$.

$\Leftarrow$ Suppose a set C in G' with $\mathbf{R}(C) \leq 2$. Denote $L_{uv} = \{u, a_{uv}, b_{uv}, c_{uv}, d_{uv}, v\}$ Note that, for any edge (u, v) oriented from u to v, $|C \cap L_{uv}| \geq 2$: if it is not the case, then there is at least one scenario inducing an evacuation path of length 3 contained in L_{uv}. In addition, $C \cap \{a_{uv}, b_{uv}, c_{uv}, d_{uv}\} \neq \emptyset$: if it was not the case, then a_{uv} for example would have an evacuation path of length 3 in scenario a_{uv} passing through vertices b_{uv}, c_{uv}, v. Therefore, $|C| = (t+|E|)$ for some non negative t. If for any edge $(u, v) \in E$ we have $|C \cap \{a_{uv}, b_{uv}, c_{uv}, d_{uv}\}| \geq 2$, then we transform C into a set $\widetilde{U}$ such that $|\widetilde{U}| \leq |C|$, $\mathbf{R}(\widetilde{U}) \leq 2$ and $|\widetilde{U} \cap \{a_{uv}, c_{uv}\}| = 1$:

- if $a_{uv} \in C$, then $\widetilde{U} = C \setminus \{b_{uv}, c_{uv}, d_{uv}\} \cup \{v\}$.
- else, if $c_{uv} \in U$, then $\widetilde{U} = C \setminus \{a_{uv}, b_{uv}, d_{uv}\} \cup \{u\}$.
- else, we have $C \cap \{a_{uv}, b_{uv}, c_{uv}, d_{uv}\} = \{b_{uv}, d_{uv}\}$, then $\widetilde{U} = C \setminus \{b_{uv}, d_{uv}\} \cup \{a_{uv}, v\}$.

It follows that, for any edge $(u, v) \in E$, at least one vertex in $\{u, v\}$ is included in $\widetilde{U}$. If $\widetilde{U}$ has less than $t + |E|$ vertices, then we add vertices in V to make $\widetilde{U}$ of cardinality $t + |E|$ without increasing any evacuation distance. In conclusion, $\widetilde{U} \cap V$ is a vertex cover of G of cardinality t, which completes the proof. □

Lemma 14 induces that the decision version of `Min Vertex Cover` in planar graphs with vertices of degree 2 or 3 polynomially reduces to $\texttt{RpCP}_2$ in bipartite planar graphs with vertices of degree 2 or 3. Since `Min Vertex Cover` is NP-complete in planar graphs with vertices of degree 2 or 3 (Lemma 7), so does $\texttt{RpCP}_2$, which concludes the proof. □

In the following we show how to adapt the proof of Proposition 13 for a stronger result. It seemed to us easier to devise directly a reduction from `Min Vertex Cover` than reducing $\texttt{RpCP}_2$ in planar graphs with vertices of degree 2 or 3 to the same problem in a more restrictive class. The proof of Proposition 13 was given only to make this reduction clearer and more intuitive.

Theorem 15. $\texttt{RpCP}_2$ is NP-complete in subgrids with vertices of degree 2 or 3, even if all edges have length 1. `RpCP` is NP-hard in this case.

Proof. We already have noted that the problem $\texttt{RpCP}_2$ is in NP. In the following we present a reduction from `Min Vertex Cover` in planar graphs of maximum degree 3. For clarity, the reduction from `Min Vertex Cover` is divided in two steps.

Step 1: This step follows general ideas proposed in [81, 79] for proving NP-hardness results in subgrids. Given a planar graph $G = (V, E)$ with vertices of degree 2 or 3, we first embed it in a grid of polynomial size using a result of [82]: vertices are mapped to vertices of the grid and edges are mapped to non-crossing paths in the grid. Embedding can be done in polynomial time. Thus, any edge (u, v) of G is replaced in the embedding by a path of length ℓ_{uv} between u and v for some positive integer ℓ_{uv}. The resulting graph is not a subgrid but only a partial subgraph of the grid. The next step will make it a subgrid with, in addition, the required properties to ensure the validity of the reduction.

Step 2: The main idea is inspired from the reduction seen in Proposition 13, where a 4-cycle is inserted on each edge of the original graph (gadget of Figure 3.1). After embedding the graph in a grid, an edge (u, v) of the original graph corresponds to a path of length ℓ_{uv} and the idea is to insert on this path a 4-cycle like in the previous reduction. The only technical difficulty is to manage the length of the paths between u, v and this 4-cycle to ensure that the reduction works. To this aim, we use the gadget T_{uv} illustrated in Figure 3.2 with vertices $x^1_{uv}, x^2_{uv}, a^+_{uv}, a^-_{uv}, b_{uv}, d_{uv}, c^+_{uv}, c^-_{uv}$ and z^i_{uv} for $i \in \{1, \ldots, 8\}$ if $\ell_{uv} = 1$, or $i \in \{1, \ldots, 9\}$ otherwise. T_{uv} is inspired from gadget L_4 in Figure 3.1. T_{uv} can replace a section of 12 consecutive horizontal edges and similarly, a sequence of 12 vertical edges can be replaced with the same gadget rotated by $\frac{\pi}{2}$ radians. To this purpose, the next step is to perform a 12-expansion on every edge (by inserting 11 new vertices). This has few advantages: it produces another embedding of the original graph in a grid that ensures that every path P_{uv} in the grid associated with an edge (u, v) of the original graph has now $12\ell_{uv}$ edges and with its 12 first edges from u (or from v) either all horizontal or all vertical. In addition, such an expansion gives enough space to ensure we can insert gadgets T_{uv} while guaranteeing the resulting graph to be a subgrid. The strategy is to insert T_{uv} by replacing the 12 first edges on one side of P_{uv}. Since T_{uv} is non symmetric, we will use an orientation of the original graph G. Starting from the graph obtained at step 1, the second step is summarized below and will conclude the reduction:

- Perform a 12-expansion. The resulting graph, G' is a subgrid obtained from G by replacing any edge (u, v) of E with a path P_{uv} of $12\ell_{uv}$ edges.
- Select an orientation of each edge of G and the related orientation on G'.
- For an edge (u, v) oriented from u to v, replace the 12 first edges of P_{uv}, starting from u, with the gadget T_{uv} represented in Figure 3.2 if the 12 replaced edges are horizontal or with the same gadget rotated by $\frac{\pi}{2}$ radians if they are all vertical.
- The resulting graph is denoted by $\widetilde{G}$

Note that for any edge (u, v) of G oriented from u to v, there is a path of length $12(\ell_{uv} - 1)$ in $\widetilde{G}$ from the vertex z^9_{uv} and v. In particular, if $\ell_{uv} = 1$, then $z^9_{uv} = v$. Else, the vertices of this path are denoted by $z^{10}_{uv}, \ldots, v = z^{9+12(\ell_{uv}-1)}_{uv}$. We then denote by $\widetilde{T}_{uv}$ the subgraph obtained by adding to T_{uv} the path from z^9_{uv} to v. If $\ell_{uv} = 1$, then $\widetilde{T}_{uv} = T_{uv}$. The graph $\widetilde{T}_{uv}$ has $6 + 12\ell_{uv}$ vertices including u and v. The graph $\widetilde{G}$ is obtained from G by replacing each edge (u, v) oriented from u to v with $\widetilde{T}_{uv}$. It is an induced subgrid since for every vertex x of $T_{uv} \setminus \{u, v\}$ and for every vertex y of $T_{u'v'} \setminus \{u', v'\}$ with $(u, v) \neq (u', v')$, (x, y) is not an edge in the grid. In addition, vertices in V have the same degree in $\widetilde{G}$ than in G and all other vertices in $\widetilde{G}$ have degree 2 or 3. $\widetilde{G}$ has $|V| + 4|E| + 12L$ vertices, all of degree 2 or 3, where $L = \sum_{(u,v)\in E} (\ell_{uv})$.

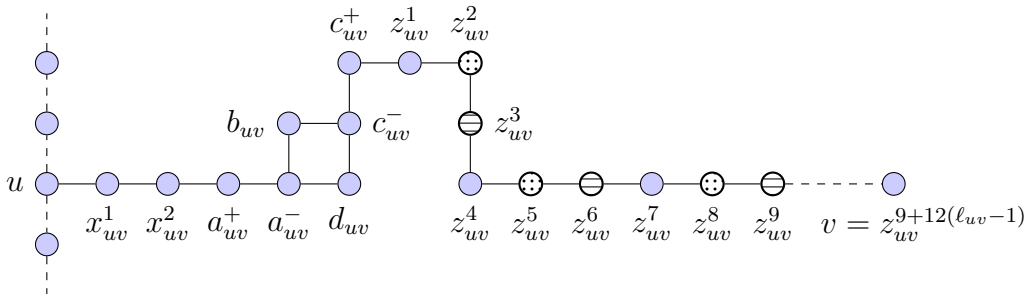


Figure 3.2: The subgraph $\widetilde{T}_{uv}$ for the edge (u, v) oriented from u to v and $\ell_{uv} > 1$. Continuous lines are used for T_{uv}'s edges while dashed lines correspond to edges outside T_{uv}.

This concludes the construction that can be performed in polynomial time. To conclude the proof of Theorem 15, we need to show that `Min Vertex Cover` in the graph G reduces to $\texttt{RpCP}_2$ in the graph $\widetilde{G}$. For this, we establish two claims. Claim 1 is a technical result used to prove Claim 2 that immediately concludes the proof.

Claim 1. Let C be a set of vertices of robust radius at most 2, then for every edge (u, v) of G oriented from u to v, the following holds:

1. C includes at least one vertex from $\{b_{uv}, d_{uv}, a^-_{uv}, c^-_{uv}\}$
2. C includes at least two vertices from $\{b_{uv}, d_{uv}, a^-_{uv}, c^-_{uv}, a^+_{uv}, c^+_{uv}\}$

Proof. (1): If $C \cap \{b_{uv}, d_{uv}, a^-_{uv}, c^-_{uv}\} = \emptyset$, then for scenario c^-_{uv}, the evacuation distance of c^-_{uv} is at least 3 (induced by the evacuation path through b_{uv} and a^-_{uv}). The same occurs for a^-_{uv}.
(2): If $b_{uv} \in C$ and $a^-_{uv} \notin C$, then d_{uv} or c^-_{uv} should be in C to ensure an evacuation

distance at most 2 for a^-_{uv} when it is on fire. So, $|C \cap \{b_{uv}, d_{uv}, a^-_{uv}, c^-_{uv}, a^+_{uv}, c^+_{uv}\}| \geq 2$. Similarly, if $d_{uv} \in C$, we have then $|C \cap \{b_{uv}, d_{uv}, a^-_{uv}, c^-_{uv}, a^+_{uv}, c^+_{uv}\}| \geq 2$. Finally, if $b_{uv}, d_{uv} \notin C$, then we need $C \cap \{a^-_{uv}, a^+_{uv}\} \neq \emptyset$ and $C \cap \{c^-_{uv}, c^+_{uv}\} \neq \emptyset$ to ensure an evacuation distance at most 2 when d_{uv} is on fire. □

We are now ready to establish Claim 2. The proof requires to define, for an edge $(u, v) \in E$ of G oriented from u to v, two disjoint sets of vertices of $\widetilde{T}_{uv}$:

Denote $C^{v+}_{uv} = \{z^{3i}_{uv}, i = 1, \ldots, 2 + 4(\ell_{uv} - 1)\}$ (last vertex is $z^{6+12(\ell_{uv}-1)}_{uv}$);

Denote $C^{v-}_{uv} = \{z^{3i-1}_{uv}, i = 1, \ldots, 3 + 4(\ell_{uv} - 1)\}$ (last vertex is $z^{8+12(\ell_{uv}-1)}_{uv}$, which is linked to v).

C^{v+}_{uv} and C^{v-}_{uv} include, after a first vertex $z^{i_0}_{uv}$, every third vertex along the path $z^i_{uv}, i_0 \leq i \leq 8 + 12(\ell_{uv} - 1)$. Roughly speaking, C^{v+}_{uv} (resp. C^{v-}_{uv}) corresponds to the optimal position of centers along the path from c^-_{uv} to v in $\widetilde{T}_{uv}$ after deciding to implement a center on vertex c^+_{uv} (resp. c^-_{uv}). Note that on a path with $3p + 1$ vertices, at least $p - 1$ centers are required in addition to the two extremities to ensure the robust radius to be at most 2. The only solution using this number of centers is to place centers at the extremities of the path and on every third vertex in between. We have $|C^{v+}_{uv}| = 2 + 4(\ell_{uv} - 1)$ and $|C^{v-}_{uv}| = |C^{v+}_{uv}| + 1$. In Figure 3.2, where $\ell_{uv} > 1$, vertices in C^{v+}_{uv} and C^{v-}_{uv} are represented striped or dashed respectively.

Claim 2. $G = (V, E)$ has a vertex cover U of size $p = |U|$ if and only if $\widetilde{G}$ has a set C_U of $(p + 4|L| + |E|)$ vertices with $\mathbf{R}(C_U) = 2$.

Proof. Let us first consider a vertex cover U of cardinality p in $G = (V, E)$. We complete U in C_U in $\widetilde{G}$ by adding, for every edge $(u, v) \in E$ of G oriented from u to v, $4\ell_{uv} + 1$ vertices as follows:

If $u \in U$, then we add to C_U vertices $\{a^+_{uv}, c^-_{uv}\} \cup C^{v-}_{uv}$.

If $u \notin U$, then we add to C_U vertices $\{x^1_{uv}, a^-_{uv}, c^+_{uv}\} \cup C^{v+}_{uv}$.

Since we add $1 + 4\ell_{uv}$ vertices for each edge, we have: $|C_U| = |U| + 4|L| + |E|$. We can check that the robust radius of C_U is at most 2 and it is at least 2 since, in all cases, we have two consecutive vertices in T_{uv} that are not in C_U.

To prove the converse, note first that we need at least $1 + 4\ell_{uv}$ vertices of $\widetilde{T}_{uv} \setminus \{u, v\}$ in C_U to ensure $\mathbf{R}(C_U) \leq 2$. Actually, if there are three consecutive vertices of degree 2 that are not in C_U, then the robust radius is at least 3. Consequently, even if u and v are in C_U, we need at least $4\ell_{uv}$ centers of $\widetilde{T}_{uv} \setminus \{u, v, b_{uv}, d_{uv}, a^-_{uv}, c^-_{uv}\}$ to ensure that every three consecutive vertices of degree 2 include at least one

center. The best way to do it is to add $\{a^+_{uv}, c^+_{uv}\} \cup C^{v+}_{uv}$, which makes $4\ell_{uv}$ centers. Using Claim 1 (first item), we deduce that at least one additional vertex from $\{b_{uv}, d_{uv}, a^-_{uv}, c^-_{uv}\}$ should be added in any set of vertices C satisfying $\mathbf{R}(C) = 2$.

Let us now assume that $\widetilde{G}$ includes a set of vertices C of robust radius at most 2. As we just noted, for every edge (u, v) of G, C includes at least $1 + 4\ell_{uv}$ vertices of $\widetilde{T}_{uv} \setminus \{u, v\}$ and thus, C includes in all $p + |E| + 4L$ vertices for some non-negative p. Suppose now that, for an edge (u, v) of G, oriented from u to v, neither u nor v is in C. Since $\mathbf{R}(C) \leq 2$, we have $\{x^1_{uv}, x^2_{uv}\} \cap C \neq \emptyset$ and $\{z^{8+12(\ell_{uv}-1)}_{uv}, z^{7+12(\ell_{uv}-1)}_{uv}\} \cap C \neq \emptyset$. Then, since we cannot have three consecutive vertices of degree 2 outside C, $|C \cap \{z^i_{uv}, i = 1, \ldots, 8 + 12(\ell_{uv} - 1)\}| \geq 4\ell_{uv} - 1$. Using Claim 1 (second item), at least two vertices from $\{b_{uv}, d_{uv}, a^-_{uv}, c^-_{uv}, a^+_{uv}, c^+_{uv}\}$ should be in C. In all, C has at least $4\ell_{uv} + 2$ vertices in $\widetilde{T}_{uv}$. These vertices can be replaced with $\{u, a^+_{uv}, c^-_{uv}\} \cup C^{v-}_{uv}$ without augmenting the cardinality of the set. By repeating this transformation, we obtain a set of vertices C' such that $|C'| \leq p + 4|L| + |E|$, where $C' \cap V$ is a vertex cover of size at most p. This completes the proof of Claim 2. □

Claim 2 states that the decision version of `Min Vertex Cover` in planar graphs with vertices of degree 2 or 3 polynomially reduces to $\texttt{RpCP}_k$ with $p = 2$ in this class of graphs. Since `Min Vertex Cover` is NP-complete in planar graphs with vertices of degree 2 or 3 (Lemma 7), so does $\texttt{RpCP}_k$. This concludes the proof of Theorem 15. □

3.1.4 Does increasing the radius make the decision problem harder?

In Subsection 3.1.1 we saw that $\texttt{RpCP}_1$ is NP-complete in all classes of graphs of minimum degree 2 in the uniform case for which the decision version of `Min Vertex Cover` problem is NP-complete. This does not include the class of subgrids in the uniform case and vertices of degree 2 or 3. Meanwhile, in Subsection 3.1.3, we proved that $\texttt{RpCP}_2$ is NP-complete in subgrids in the uniform case and vertices of degree 2 or 3.

These results make natural the question of the hardness of $\texttt{RpCP}_k$ when k varies. We can easily see that $\texttt{RpCP}_k$ for $k > 1$ is NP-complete for classes of graph which are *stable by transitivity* and for which $\texttt{RpCP}_1$ is NP-complete: A class $\mathcal{H}$ is stable by transitivity if and only if, for a graph $G = (V, E) \in \mathcal{H}$, the graph $G' = (V, E^k)$ - obtained from G such that $E^k = \{(i, j) : \forall i, j \in V, d(i, j) \leq k, \}$ - is in $\mathcal{H}$. In this case, the instance (G, p) of $\texttt{RpCP}_k$ can be reduced to the instance (G', p) of $\texttt{RpCP}_1$.

However, a natural question is whether there exists a reduction allowing to state for a given k the hardness of $RpCP_{k+1}$ on a class if $\texttt{RpCP}_k$ is known to be hard on this class? In particular, can we conclude results for larger values of k for the classes studied here? The following remark gives evidence of graph classes on

which $\texttt{RpCP}_1$ is hard but $\texttt{RpCP}_2$ is trivial. This justifies that such a reduction from k to $k+1$ does not exist in the general case and consequently, the hardness of $\texttt{RpCP}_k$ on a given graph class requires to be studied for any value of k and cannot be deduced, in general, from hardness results dealing with different values of k. This leaves open avenues for future researches with, in particular, the challenge to devise reductions working for different values of k.

Remark 2. There are graph classes for which $\texttt{RpCP}_1$ is NP-complete while all graphs in the class have a solution of robust radius 2 using only two centers (thus, $\texttt{RpCP}_2$ is trivial).

Proof. Consider any class $\mathcal{H}$ of graphs with all vertices of degree at least 2 and including at least three independent vertices (i.e., not linked by an edge) such that the decision version of `Min Vertex Cover` problem is NP-complete on $\mathcal{H}$. The condition that at least three independent vertices exist in any graph $G = (V, E)$ of this class is not restrictive since `Min Vertex Cover` can be trivially solved in polynomial time on graphs that do not satisfy this condition. The condition ensures that the size of a `Min Vertex Cover` is at most $|V|-3$ for any graph $G = (V, E) \in \mathcal{H}$. We build the class $\mathcal{H}'$ of all graphs obtained from a graph $G \in \mathcal{H}$ by adding two vertices u_0, v_0 completely linked to all vertices of G. For any graph in $\mathcal{H}'$, a `Min Vertex Cover` includes u_0, v_0 and a `Min Vertex Cover` of the graph obtained by removing u_0, v_0.

If u_0 or v_0 is not included in the vertex cover, then all other vertices should be included which, by hypothesis, is larger than the proposed solution. Therefore, the decision version of `Min Vertex Cover` is NP-complete on $\mathcal{H}'$, and using Corollary 9 so is $\texttt{RpCP}_1$. Note however that, for any graph in $\mathcal{H}'$ obtained from $G = (V, E) \in \mathcal{H}$ by adding u_0, v_0, the set $\{u_0, v_0\}$ is a solution of RpCP of robust radius 2 with $p = 2$ not possible with $p = 1$, which completes the proof. □

3.2 PpCP Non-Approximation Results

We present in this section two non-approximation results for PpCP. The first one, on bipartite planar graphs of degrees 2 or 3, shows that PpCP can not be approximated with a ratio less than $\frac{20}{19}$ (Section 3.2.1). The second one, on subgrids of degrees 2 or 3, shows that PpCP can not be approximated with a ratio less than $\frac{56}{55}$ (Section 3.2.2). We first establish a technical lemma about vertex cover in graphs.

Lemma 16. Let $G = (V, E)$ be a graph and $G' = (V', E')$ be the graph obtained by inserting $2k_{uv}$ vertices on each edge $(u, v) \in E$, where k_{uv} is a non-negative integer. Then we have

$$\tau(G') = \tau(G) + \sum_{uv \in E} k_{uv}$$

Npte that, if for some integer k we have $\forall (u, v) \in E, k_{uv} = k$, then G' is a $(2k+1)$-expansion of G.

Proof. For every edge $(u, v) \in E$ oriented from u to v, we denote $P_{uv} = G'[\{u, x_{uv}^1, \ldots, x_{uv}^{2k_{uv}}, v\}]$ the path between u and v in G'. Note that at least k_{uv} vertices are needed to cover the edges in $P_{uv} \setminus \{u, v\}$.

Let $U \subset V$ a vertex cover of G: $\forall (u, v) \in E, \{u, v\} \cap U \neq \emptyset$. We can build $U' \subset V'$ in G' as follows. We initialize U' with all vertices of U. Then, for every edge $(u, v) \in E$, if $u \in U$, we add vertices $x_{uv}^{2i}, 1 \leq i \leq k_{uv}$ to U'. Otherwise, $v \in U$ necessarily, then we add vertices $x_{uv}^{2i+1}, 0 \leq i \leq k_{uv} - 1$ to U'. In both cases we have added exactly k_{uv} vertices and all edges of $P_{uv}^{G'}$ are covered by U', with $|U'| = |U| + \sum_{uv \in E} k_{uv}$. Then $\tau(G') \leq \tau(G) + k_{uv}$.

Assume now that G' has a vertex cover X'. For every $(u, v) \in E$, $P_{uv}^{G'}$ is covered by at least $k_{uv} + 1$ vertices. If $u, v \notin X'$, we can transform X' into U' such that u or v is in U'. Then $|U' \setminus V| \geq \sum_{uv \in E} k_{uv}$. Since at least one vertex between u and v is in $U = V \cap U'$, U is a vertex cover for G. Then $|U| = |U'| - k_{uv}$, thus $\tau(G) \leq \tau(G') - k_{uv}$. Hence $\tau(G') = \tau(G) + k_{uv}$ and the proof is complete. □

3.2.1 PpCP Non-Approximable on Planar Graphs

In this section, we give an non-approximability result for PpCP on bipartite planar graphs of maximum degree 3. The case of planar graphs, in particular with low degree, is natural for our application. It motivates us to investigate the complexity status of our problem in restricted classes of planar graphs to better discriminate polynomial cases and hard cases.

For our first transformation, we use the same gadget as in Subsection 3.1.3. Given a planar graph $G = (V, E)$ without pending vertex, one builds a bipartite graph $G' = (V', E')$ by replacing each edge uv by the gadget L_4 as presented in Figure 3.1. We have seen in Lemma 14 that G has a vertex cover of size t if and

only if G' has a set C' of size $t + |E|$ with $\mathbf{R}(C) \leq 2$. Moreover, for each edge $(u, v) \in E$, C' has exactly one vertex in $\{a_{uv}, c_{uv}\}$ and none in $\{b_{uv}, d_{uv}\}$. Given Proposition 11, we can conclude that C' is a dominating set of G'.

Assume G has a minimal[2] vertex cover of size t, we define $p_t = t + |E|$ and consider a dominating set C' of size p_t in G', with exactly one vertex in $\{a_{uv}, c_{uv}\}$ and none in $\{b_{uv}, d_{uv}\}$ for any edge $uv \in E$. We evaluate the evacuation radius of C' in G' for any scenario $s \in V'$. We recall that $V' \subset V$.

Lemma 17. Using the above notations, we have:

$$r^s(C') = \begin{cases} 1 & \Leftrightarrow \quad s \in V \setminus C' \\ 2 & \Leftrightarrow \quad s \in V \cap C' \text{ and } s \notin V \end{cases}$$

Proof. Since G' is triangle-free with no pending vertex, and C' is a dominating set, then we have: $\forall u \in V', r^s(C', u) \leq 2$ for any scenario s (Proposition 11).

Consider first $s \in V \setminus C'$ and denote by $v_1, \ldots, v_{deg(s)}$ all neighbors of s in G. By definition of a dominating set, $\{v_1, \ldots, v_{deg(s)}\} \subset C'$. Consequently, by construction of C', we have that $\{a_{sv_1}, \ldots, a_{sv_{deg(s)}}\} \subset C'$. It follows that, whatever the escaping direction, people located on s will reach a center at distance 1, which gives $r^s(C', s) = 1$. In addition, since C' is a dominating set in G', every other vertex u of the graph is adjacent to at least one element of C', which gives $r^s(C', u) = 1$. Thus, $r^s(C') = 1$.

Consider now $s \in V \cap C'$. Since the considered vertex cover is minimal, there is $j \in \{1, \ldots, deg(s)\}$ such that $v_j \notin C'$ and thus $C' \cap \{a_{sv_j}, b_{sv_j}, c_{sv_j}, d_{sv_j}\} = \{c_{sv_j}\}$. Then $\{a_{sv_j}, b_{sv_j}, c_{sv_j}\}$ is an evacuation path of a_{sv_j}. It follows that the evacuation distance of a_{sv_j} is 2, thus $r^s(C') = 2$.

Consider now $s \in \{b_{uv}, d_{uv}\}$ for $uv \in E$. Consequently, by construction of C', we have that $s \notin C'$ and that only one neighbor of s is in C'. Then s has an evacuation path that crosses its neighbor vertex not in C', which means the evacuation distance of s is 2. Thus $r^s(C') = 2$.

Finally, consider $s \in \{a_{uv}, c_{uv}\}$ for $uv \in E$. Suppose $s = a_{uv}$. If $s \notin C'$, then $c_{uv} \in C'$ and the evacuation distance of s is 2. Otherwise, that is if $s \in C'$, then the evacuation distance of b_{uv} and d_{uv} is 2. We can proceed similarly with the case $s = c_{uv}$, thus $r^s(C') = 2$

In conclusion, for all cases except the first one, $r^s(C') = 2$. Thus the proof is complete. □

At last, we will need the following lemma for our theorem.

Lemma 18. For $G = (V, E)$ a graph with degrees $\{2, 3\}$ and $p < \gamma(G)$, the minimum probabilistic radius of any solution of PpCP is greater than 2.

[2]A set C is minimal for inclusion with respect to a property π if the removal of an element of C causes this set to no longer verify the property π.

Proof. Let C be a feasible solution of PpCP on G for $p < \gamma(G)$. As C cannot be a dominating set, there exists $v \in V$ such that $(\{v\} \cup \Gamma(v)) \cap C = \emptyset$ i.e., v is not in C or adjacent to any vertex of C. For any scenario s, the evacuation distance of v will be at least 2 as none of its neighboring vertices is in C. Thus $r^s(C, v) \geq 2, \forall s \in V$, which implies $r^s(C) \geq 2, \forall s \in V$. In addition, for any vertex $y \in \Gamma(v)$, the evacuation distance of y in scenario y is at least 3 since y has an evacuation path that crosses v. It follows that $r^y(C) \geq 3$. Combined to $r^s(C, v) \geq 2, \forall s \in V$, we get $\mathbf{E}(C) > 2$ for any feasible solution C of PpCP on G. □

We are now ready to prove the main result of this section.

Theorem 19. There is no polynomial time approximation for PpCP guaranteeing a ratio less than $\frac{20}{19}$ for bipartite planar graphs of degrees 2 or 3, unless P=NP.

Proof. The proof is by contradiction. Let ρ satisfy $1 < \rho < \frac{20}{19}$. Consider $\varepsilon > 0$ such that $\rho < \frac{20+2\varepsilon}{19+2\varepsilon} < \frac{20}{19}$. Take $q \in \mathbb{N}$, such that $\frac{5}{q} \leq \varepsilon$ and $q \geq 2$.

We suppose we have a polynomial approximation algorithm A for PpCP, admitting as argument a planar graph $\mathcal{P}$ of degrees 2 or 3 and the number p of centers, and guaranteeing the approximation ratio ρ. We will show how to use this algorithm to solve the Min Vertex Cover problem on planar graphs with vertex degrees 2 or 3. Lemma 7 will give the contradiction, unless P=NP.

Consider a planar graph $G = (V, E)$ with vertex degrees in $\{2, 3\}$. We perform first a $2q + 1$-expansion on G and denote $G_q = (V_q, E_q)$ the resulting graph. Next we transform G_q in the graph $G'_q = (V'_q, E'_q)$ by inserting the gadget L_4 as presented in Figure 3.1. We have in particular:

$$\begin{array}{rcl} |V_q| & = & |V| + 2q|E| \\ |E_q| & = & (2q+1)|E| \\ |V'_q| & = & |V_q| + 4|E_q| \\ & = & |V| + (10q+4)|E| \end{array} \tag{3.1}$$

From Lemma 16 we have $\tau(G_q) = \tau(G) + q|E|$ and from Proposition 11 and Lemma 14 we deduce $\gamma(G'_q) = \tau(G_q) + |E_q|$. Then we obtain:

$$\gamma(G'_q) = \tau(G) + (3q+1)|E| \tag{3.2}$$

We apply the hypothetical approximation algorithm A on G'_q for different values of p, starting with $p = 1$ and augmenting it. While $p < \gamma(G'_q)$, the algorithm computes a solution C of probabilistic radius strictly greater than 2 (Lemma 18). Suppose now we use $p = \gamma(G'_q) = \tau(G_q) + |E_q|$. Then using Lemma 14, there is a p-set C of graph G'_q satisfying the conditions of Lemma 17. Thus we obtain:

$$|V'_q|\mathbf{E}(C) = 2|V'_q| - (|V_q| - \tau(G_q))$$

We deduce, using Relations 3.1 and 3.2:

$$\begin{aligned} |V'_q|\mathbf{E}(C) &= |V| + (19q+8)|E| + \tau(G) \\ &< 2|V| + (19q+8)|E| \end{aligned}$$

where the last inequality holds because $\tau(G) < |V|$. So, we have:

$$\mathbf{E}(C) < \frac{2|V| + (19q+8)|E|}{|V| + (10q+4)|E|} = 2 - \frac{q|E|}{|V| + (10q+4)|E|}$$

Since G has vertices of degree at least 2, we have $V \leq |E|$, thus:

$$\mathbf{E}(C) < 2 - \frac{q|E|}{(10q+5)|E|} = 2 - \frac{q}{10q+5} \leq \frac{19+2\varepsilon}{10+\varepsilon}$$

where the last inequality holds since $\frac{5}{q} \leq \varepsilon$. As a consequence, since an optimal solution C^* will satisfy $\mathbf{E}(C^*) \leq \mathbf{E}(C)$, the approximation algorithm `A` will determine an approximated center C in $G_q{}'$ of value:

$$\begin{aligned} \mathbf{E}(C) &\leq p - 1 \times \tfrac{19+2\varepsilon}{10+\varepsilon} \\ &< \tfrac{20+2\varepsilon}{19+2\varepsilon} \times \tfrac{19+2\varepsilon}{10+\varepsilon} \\ &= 2 \end{aligned} \tag{3.3}$$

Since constructing G'_q and evaluating $\mathbf{E}(C)$ can be done in polynomial time, and since algorithm `A` will be run less than $|V|$ times, the whole process is polynomial. If P≠NP, then this is a contradiction, and the proof is complete. □

3.2.2 `PpCP` Inapproximation subgrids

In this section, we prove that `PpCP` cannot be approximated with a ratio less than $\frac{56}{55}$ on a restricted subclass of bipartite planar graphs, the class of subgrids of degrees $\{2,3\}$. This section is dedicated to prove Theorem 22. In Subsection 3.2.2.1 we explain the general scheme of the demonstration before giving all details in Subsections 3.2.2.2, 3.2.2.3 and 3.2.2.4.

3.2.2.1 Global blueprint of the proof

In Theorem 22, we will show that a polynomial time approximation algorithm `A` for `PpCP` in subgrid of degrees $\{2,3\}$ guaranteeing a ratio of at most $\frac{56}{55}$ could be used to compute in polynomial time the size of the minimum vertex cover on a planar graph of degrees $\{2,3\}$, which is a contradiction.

We start from a planar graph $G = (V, E)$ with degrees $\{2,3\}$, instance of `Min Vertex Cover`. We randomly choose an orientation of the edges of G that will be

used in our reductions and analysis. We then apply successively two transformations, Transformation 1, denoted φ_1 and Transformation 2, denoted φ_2 that are detailed in Subsection 3.2.2.2. Figure 3.3 gives a simple schematic representation of the whole reduction. Note that the subgrid H appearing in Figure 3.3 is an intermediate stage not directly used in the analysis.

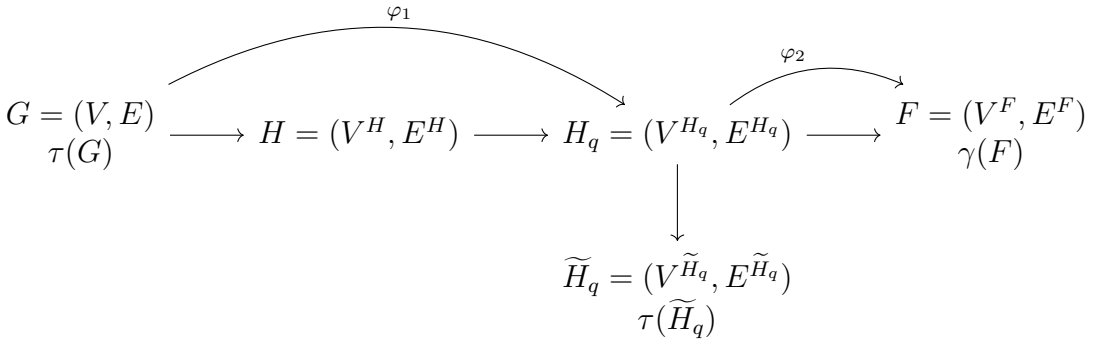


Figure 3.3: The different graphs involved in the reduction.

Transformation 1 (φ_1) constructs from G a subgrid $H_q = (V^{H_q}, E^{H_q})$, for some positive integer q specified later, in such a way that:

- $V \subset V^{H_q}$,
- Edges (u, v) of G map to non-crossing paths $P_{uv}^{H_q}$ of even length between u and v in H_q.

We then apply Transformation 2 (φ_2) to construct a subgrid $F = (V^F, E^F)$ from H_q. Roughly speaking, it consists in replacing the first two edges of $P_{uv}^{H_q}$ (where $(u, v) \in E$ is directed from u to v) with a gadget T^2, and every other edge of H_q with a gadget T^1, both defined in the next subsection.

At this point, note that there is no direct and easy link between $\tau(G)$ and $\tau(H_q)$ since $\tau(H_q)$ can be obtained in polynomial time (H_q is bipartite) while G is meant to be an instance of an NP-hard restriction of `Min Vertex Cover`. For this reason, we introduce an auxiliary graph $\widetilde{H}_q = (V^{\widetilde{H}_q}, E^{\widetilde{H}_q})$. It can be seen as a perturbation of H_q with a direct link between $\tau(\widetilde{H}_q)$ and $\tau(G)$. It is simply obtained by replacing, for every edge $(u, v) \in E$, the two first edges of the path $P_{uv}^{H_q}$ by a single edge. This way, the path $P_{uv}^{H_q}$ of even length becomes, in $\widetilde{H}_q$, a path $P_{uv}^{\widetilde{H}_q}$ of odd length and Lemma 16 can be used to write $\tau(\widetilde{H}_q)$ as a function of $\tau(G)$. On the other hand, as outlined in Lemma 20, the properties of the two gadgets allow to establish a direct link between dominating sets in F and vertex covers in $\widetilde{H}_q$. In all, it gives a relation between the `Min Dominating Set` problem in F and the `Min Vertex Cover` problem in G.

Then, in Subsection 3.2.2.3, we outline different relations between the `Min Dominating Set` problem and `PpCP` in a triangle-free graphs without pending vertices using Lemmas 11 and 20. This can be applied to F.

Finally, in Subsection 3.2.2.4, we use these results to establish Theorem 22. We show that, when applying `A` on F for $p < \gamma(F)$, the output is a solution of `PpCP` of expected radius at least 2, while applying it for $p = \gamma(F)$ gives a solution of expected radius less than 2. Hereby we can use such an algorithm to compute $\gamma(F)$, and consequently $\tau(G)$. Since constructing H, H_q, $\widetilde{H}_q$ and F, as well as evaluating the value of a `PpCP` solution, can be done in polynomial time, and since algorithm `A` is applied less than $|V|$ times, the whole process is polynomial.

3.2.2.2 Details on the transformations and their properties

Transformation 1. From a planar graph $G = (V, E)$ to a subgrid $H_q = (V^{H_q}, E^{H_q})$ with $q > 0$.

Using a result of [82], we can embed $G = (V, E)$ in a grid $H = (V^H, E^H)$ of polynomial size. Vertices of G are mapped to vertices of the grid, and edges (u, v) of G map to non-crossing paths P_{uv}^H between u and v in the grid. Note that we cannot control the length and parity of these paths. The resulting graph is a partial subgrid and not necessarily a subgrid yet. We then perform a $2q$-expansion for some positive integer q specified later. The resulting graph $H_q = (V^{H_q}, E^{H_q})$ is a subgrid ($q > 0$). In addition, since the expansion multiplies by $2q$ all path lengths from H to H_q, edges (u, v) of G map to non-crossing paths $P_{uv}^{H_q}$ of even length between u and v in H_q. It means that paths $P_{uv}^{H_q}$ have $2k_{uv} + 1$ internal vertices (excluding u and v) for some non-negative integers k_{uv}.

Example. Suppose the planar graph $G = (V, E)$ is a complete graph on four vertices $\{a, b, c, d\}$ as presented in Figure 3.4 and set $q = 2$. We choose an orientation of G such that the directed edges of G are $\{(a, b), (a, c), (b, c), (c, d)\}$. $H = (V^H, E^H)$ corresponds to a possible embedding of G in a grid, where the edge $(a, d) \in E$ maps to the path $\{a, z_{ad}, d\}$ in H. Next, we construct the subgrid H_q by applying the $2q$-expansion. The resulting graph H_q can be seen on the right side of Figure 3.4. Finally, the related graph $\widetilde{H}_q$ is represented in Figure 3.5. ♣

Analysis of φ_1: As already noticed in Subsection 3.2.2.1, we cannot establish a direct link between $\tau(G)$ and $\tau(H_q)$. However, since we now control the parity of paths $P_{uv}^{H_q}$, it is easy to slightly modify H_q so as we can apply Lemma 16. This is the role of the graph $\widetilde{H}_q = (V^{\widetilde{H}_q}, E^{\widetilde{H}_q})$. Recall that this graph is obtained from H_q by replacing, for every edge $(u, v) \in E$, the two first edges of the path $P_{uv}^{H_q}$ by a single edge, as illustrated in Figure 3.5. This way, $\widetilde{H}_q$ can directly be obtained

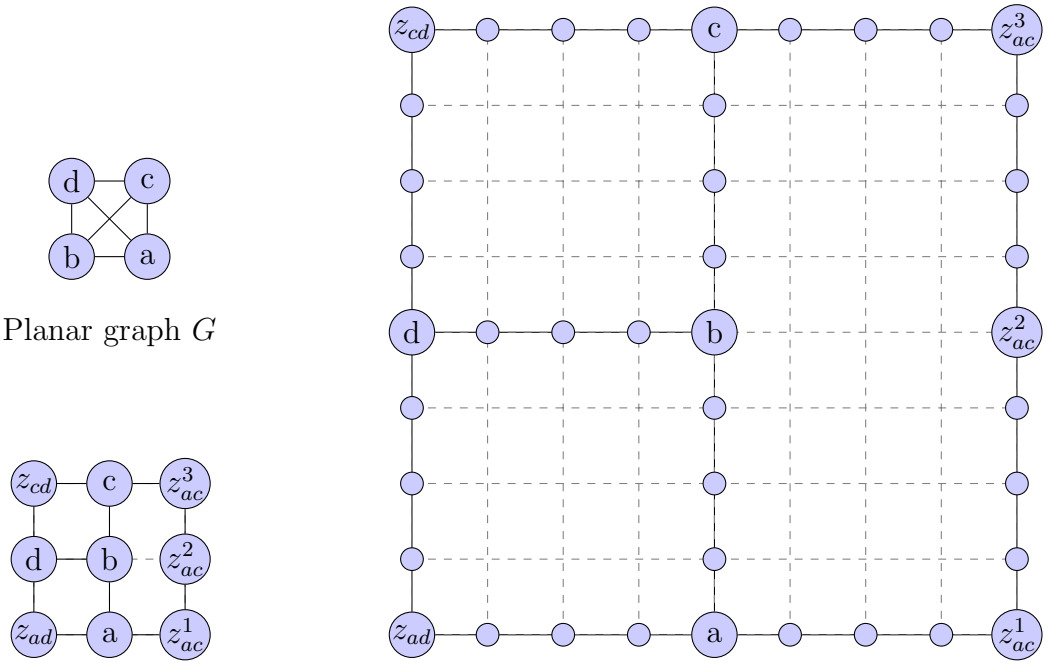


Figure 3.4: Example of Transformation 1

from G by inserting $2k_{uv}$ vertices on each edge $(u, v) \in E$. As a consequence, by Lemma 16, we have:

$$\tau(\widetilde{H}_q) = \tau(G) + \sum_{(u,v)\in E} k_{uv}. \tag{3.4}$$

In addition, we have:

$$\begin{aligned} |V^{\widetilde{H}_q}| &= |V| + 2\sum_{e\in E} k_e \\ |E^{\widetilde{H}_q}| &= |E| + 2\sum_{e\in E} k_e \end{aligned} \tag{3.5}$$

By construction, we have $\forall (u, v) \in E, 2k_{uv} + 1 \geq 2q - 1$, which gives:

$$\forall (u, v) \in E, k_{uv} \geq q - 1. \tag{3.6}$$

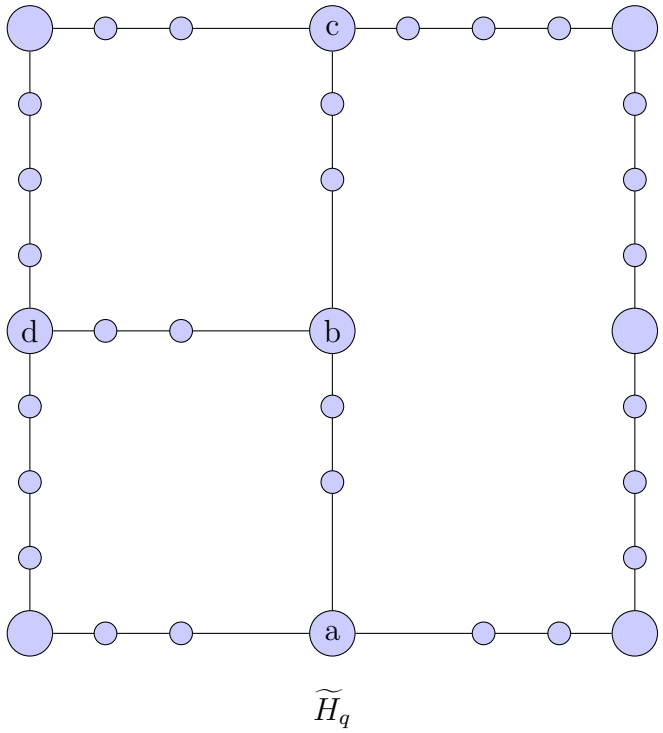


Figure 3.5: The graph $\widetilde{H}_q$ obtained from G through H_q.
$k_{ab} = k_{bc} = k_{bd} = 1$, $k_{ad} = k_{cb} = 3$ and $k_{ac} = 7$

Transformation 2. From subgrid $H_q = (V^{H_q}, E^{H_q})$ to subgrid $F = (V^F, E^F)$.

Thanks to the $2q$-expansion, for $(u, v) \in E$ directed from u to v, the first two edges of $P_{uv}^{H_q}$ in H_q are both horizontal or vertical. Note as well that the orientation of G immediately defines an orientation of H_q and of $\widetilde{H}_q$. We can then construct the subgrid $F = (V^F, E^F)$ from the subgrid H_q as follows.

For every edge $(u, v) \in E$ directed from u to v, we replace, in H_q, the first two edges $(u, i), (i, x)$ of $P_{uv}^{H_q}$ with T_{ux}^2 defined in Figure 3.7, and every other edges $(x, y) \in E^{H_q}$ with T_{xy}^1 defined in Figure 3.6.

In the following we use T_{xy}^{12} to refer to T_{xy}^1 or T_{xy}^2. Note that two gadgets T_{xy}^{12} never overlap each other in F and the resulting graph F is a subgrid. Indeed, if G is embedded in a grid $\mathcal{G}$, H_q is embedded in $Exp(\mathcal{G}, 2q)$ and F is embedded in $Exp(\mathcal{G}, 14q)$.

Analysis of φ_2: By construction we have $|V^F| = |V^{\widetilde{H}_q}| + 13|E^{\widetilde{H}_q}| + 3|E|$ and $|E^F| = 15|E^{\widetilde{H}_q}| + 3|E|$. Using Relation 3.5, we deduce:

$$\begin{aligned} |V^F| &= |V| + 16|E| + 28 \sum_{e \in E} k_e \\ |E^F| &= 18|E| + 30 \sum_{e \in E} k_e \end{aligned} \tag{3.7}$$

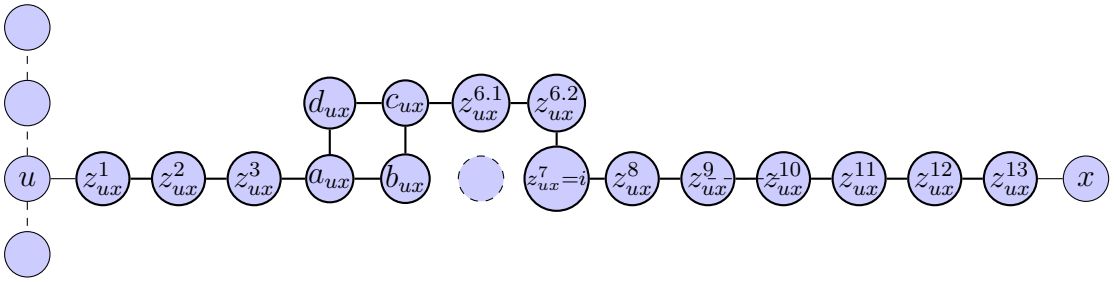


Figure 3.6: Gadget T^2_{ux} used in F for $(u,i),(i,x) \in E^{H_q}$; $z^7_{ux} = i$.

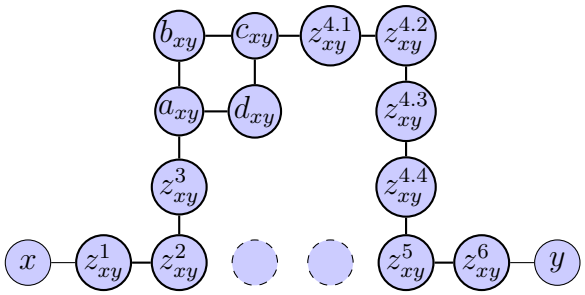


Figure 3.7: Gadget T^1_{xy} used in F for $(x,y) \in E^{H_q}$.

Lemma 20. For any $t \leq |V|$, $\widetilde{H}_q = (V^{\widetilde{H}_q}, E^{\widetilde{H}_q})$ has a vertex cover of size t if and only if F has a dominating set D of size $t + 4|E^{\widetilde{H}_q}| + |E|$ such that, for each edge $(x,y) \in E^{\widetilde{H}_q}$, we have:

- at least one vertex of $\{a_{xy}, c_{xy}\}$ is in D
- at least one vertex of $\{z^1_{xy}, z^{13}_{xy}\}$ if (x,y) is the first edge of a path $P^{\widetilde{H}_q}_{uv}$ with $(u,v) \in E$ directed from u to v
- at least one vertex in $\{z^1_{xy}, z^6_{xy}\}$ in the other cases.

Proof. For this result, it is convenient to see how F could be constructed from $\widetilde{H}_q$: for every edge $(u,v) \in E$ directed from u to v, the first edge of $P^{\widetilde{H}_q}_{uv}$ - we denote $E^{\widetilde{H}_q}_2$ the set of such edges corresponding to two edges of $P^{H_q}_{uv}$ - is replaced with

T^2_{uv}. All other edges of $\widetilde{H_q}$ - we denote $E_1^{\widetilde{H_q}} \subset E^{\widetilde{H_q}}$ their set - are replaced with T^1_{uv}. Note that $|E_2^{\widetilde{H_q}}| = |E|$.

$\Rightarrow$ Let $U \subset V^{\widetilde{H_q}}$ be a vertex cover of $\widetilde{H_q}$ of size t. We initialize D with all vertices of U, seen as a subset of V^F, and complete it in a dominating set of F. Then for every $(x, y) \in E^{\widetilde{H_q}}$, oriented from x to y, we have $D \cap \{x, y\} \neq \emptyset$. We then apply one of the two following cases:

- if $(x, y) \in E_2^{\widetilde{H_q}}$: If $x \in D$, we add to D the vertices $z^3_{xy}, c_{xy}, z^7_{xy}, z^{10}_{xy}$ and z^{13}_{xy} of T^2_{xy}, else if $y \in D$, we add to D the vertices $z^1_{xy}, a_{xy}, z^{6.1}_{xy}, z^8_{xy}$ and z^{11}_{xy} of T^2_{xy}. In both cases, 5 vertices are added to D, and all the vertices of T^2_{xy} are dominated by D.

- if $(x, y) \in E_1^{\widetilde{H_q}}$: If $x \in D$, we add to D the vertices $z^3_{xy}, c_{xy}, z^{4.3}_{xy}$ and z^6_{xy} of T^1_{xy}, else if $y \in D$, we add to D the vertices $z^1_{xy}, a_{xy}, z^{4.1}_{xy}$ and $z^{4.4}_{xy}$ of T^1_{xy}. In both cases, 4 vertices are added to D, and all the vertices of T^1_{xy} are dominated by D.

The resulting set D is a dominating set of F of size $t + 4|E^{\widetilde{H_q}}| + |E_2^{\widetilde{H_q}}| = t + 4|E^{\widetilde{H_q}}| + |E|$ and for each edge $(x, y) \in E^{\widetilde{H_q}}$, D has at least one vertex in $\{a_{xy}, c_{xy}\}$ and one vertex in $\{z^1_{xy}, z^{13}_{xy}\}$ (resp. $\{z^1_{xy}, z^6_{xy}\}$) if $(x, y) \in E_2^{\widetilde{H_q}}$(resp. $E_1^{\widetilde{H_q}}$).

$\Leftarrow$ Now suppose we have D a dominating set of F. Then for every $(x, y) \in E^{\widetilde{H_q}}$ oriented from x to y, we have:

- if $(x, y) \in E_2^{\widetilde{H_q}}$: D includes at least 6 vertices on T^2_{xy}, and 5 vertices on $T^2_{xy} \setminus \{x, y\}$.

- if $(x, y) \in E_1^{\widetilde{H_q}}$: D includes at least 5 vertices on T^1_{xy}, and 4 vertices on $T^1_{xy} \setminus \{x, y\}$.

Then D includes at least $t' + 4|E^{\widetilde{H_q}}| + |E_2^{\widetilde{H_q}}| = t' + 4|E^{\widetilde{H_q}}| + |E|$ vertices for some integer t'. We then perform the following modifications on D:

- for every $(x, y) \in E_2^{\widetilde{H_q}}$ oriented from x to y : if $x \in D$, we can replace at least 5 vertices of $D \cap T^2_{xy}$ by $z^3_{xy}, c_{xy}, z^7_{xy}, z^{10}_{xy}$ and z^{13}_{xy}. If $y \in D$, we can replace at least 5 vertices of $D \cap T^2_{xy}$ by $z^1_{xy}, a_{xy}, z^{6.1}_{xy}, z^8_{xy}$ and z^{11}_{xy}.

- for every $(x, y) \in E_1^{\widetilde{H_q}}$ oriented from x to y : if $x \in D$, we replace at least 4 vertices of $D \cap T^1_{xy}$ by $z^3_{xy}, c_{xy}, z^{4.3}_{xy}$ and z^6_{xy}. If $y \in D$, we replace at least 4 vertices of $D \cap T^1_{xy}$ by $z^1_{xy}, a_{xy}, z^{4.1}_{xy}$ and $z^{4.4}_{xy}$. If neither x nor y is in D, we can induce that $|D \cap T^1_{xy}| \geq 5$. Thus, we replace at least 5 vertices of $D \cap T^1_{xy}$ by $x, z^3_{xy}, c_{xy}, z^{4.3}_{xy}$ and z^6_{xy}.

Note that none of these modifications increases the size of D, and D is still a dominating set of F. However, we ensured that $|D \setminus V^{\widetilde{H}_q}| \geq 4|E| + |E|$, and $|D \cap \{x, y\}| \geq 1, \forall (x, y) \in E^{\widetilde{H}_q}$. Then $U = D \cap V^{\widetilde{H}_q}$ is a vertex cover for $\widetilde{H}_q$ of size at least t. This completes the proof. □

3.2.2.3 Relations between PpCP and dominating sets

Lemma 21. Let D be a minimum dominating set of F as described in Lemma 20 and of size $p_t = \tau(G) + 4|E^{\widetilde{H}_q}| + |E_2^{\widetilde{H}_q}|$. Then we have:

$$r^s(D) = \begin{cases} 1 & \text{if } s \in V^{\widetilde{H}_q} \subset V^F \text{ and } s \notin D \\ 2 & \text{otherwise} \end{cases}$$

Proof. Since D is a dominating set of the triangle-free F, by Lemma 11 D is a feasible solution of PpCP for $p = p_t$ such that $r^s(D, v) \leq 2, \forall s, v \in V^F$. We recall that every vertex of $\widetilde{H}_q$ maps a vertex in F by construction, thus we consider $V^{\widetilde{H}_q} \subset V^F$ in the following. Three cases emerge:

1. $s \in V^F \setminus V^{\widetilde{H}_q}$: Denote $(x, y) \in E^{\widetilde{H}_q}$ such that $s \in T_{xy}^{12}$. As D is a minimal dominating set of F, D is build as the resulting dominating set described in Lemma 20. It follows that there is at least one evacuation distance of length 2 for any scenario $s \in V^F \setminus V^{\widetilde{H}_q}$, i.e $r^s(D) = 2$.

In the following, $s \in V^{\widetilde{H}_q}$ and we denote by $u_1, \ldots, u_d \in V^{\widetilde{H}_q}$ the neighbors of s in $\widetilde{H}_q$.

2 $s \in V^{\widetilde{H}_q} \cap D$: Since D is minimal, $D \cap V^{\widetilde{H}_q}$ is a minimal vertex cover of $\widetilde{H}_q$, thus there is at least one neighbor $u \in \{u_1, \ldots, u_d\}$ of s in $\widetilde{H}_q$ that is not included in D. By construction, $z_{su}^1, z_{su}^2 \notin D$ and $z_{su}^3 \in D$. Then under scenario s, the evacuation distance of z_{su}^1 is 2, i.e. $r^s(D, z_{su}^1) = 2$. Under scenario s, the evacuation distance of any other vertex in T_{su} is less than 2 given that D is a minimal dominating set. For any other neighbor $u' \in \{u_1, \ldots, u_d\}$ of s in $\widetilde{H}_q$ ($u' \neq u$), we have $|\{z_{su'}^1, z_{su'}^2, z_{su'}^3\} \cap D| = 1$, and D a minimal dominating set on $T_{su'}$, thus the evacuation distance of any vertex in $T_{su'}$ is at most 2. Therefore $r^s(D) = 2$.

3 $s \in V^{\widetilde{H}_q} \setminus D$: We recall that by definition $D \cap V^{\widetilde{H}_q}$ is a minimal vertex cover of $\widetilde{H}_q$, then $\{u_1, \ldots, u_d\} \subset D$. In addition, for any edge $(s, u) \in E^{\widetilde{H}_q}$ oriented from s to u, D includes by construction z_{su}^1. Then every neighbor of s in F is included in D by construction. Therefore, $r^s(D, s) = 1$. Since D is a dominating set in F, it remains a dominating set in $F - s$, which guarantees $r^s(D, v) = 1, \forall v \in V^F - s$. Thus $r^s(D) = 1$.

So, in all cases except the last one, $r^s(U) = 2$, and the proof is complete. □

We now are ready to prove the main result of this section.

3.2.2.4 The theorem

Theorem 22. If P≠ NP, there is no polynomial time approximation for `PpCP` guaranteeing a ratio less than $\frac{56}{55}$ for subgrids with vertex degrees 2 or 3, even in the uniform case.

Proof. The proof is by contradiction. Let us suppose there is a polynomial approximation algorithm `A` for uniform `PpCP` which guarantees the approximation ratio ρ satisfying $1 < \rho < \frac{56}{55}$, on subgrids with vertex degrees 2 or 3 for a parameter p. We will show how to use this algorithm to solve the `Min Vertex Cover` problem on planar graphs. Lemma 7 gives the contradiction, unless P=NP.

Suppose $\varepsilon > 0$ such that $\rho < \frac{56+2\varepsilon}{55+2\varepsilon} < \frac{56}{55}$. Take an integer $q \geq 2$ such that $\varepsilon \geq \frac{17}{q-1}$.

Consider a planar graph $G = (V, E)$, instance of `Min Vertex Cover`. Consider the graph H_q obtained by Transformation 1, as well as $\widetilde{H}_q = (V^{\widetilde{H}_q}, E^{\widetilde{H}_q})$ and the vector $\{k_e : e \in E\}$ obtained through H_q. In addition, consider the graph $F = (V^F, E^F)$ obtained from H_q through Transformation 2.

Recall that, from Relations 3.5 and 3.7, we have $|V^{\widetilde{H}_q}| = |V| + 2\sum_{e\in E} k_e$, $|E^{\widetilde{H}_q}| = |E| + 2\sum_{e\in E} k_e$ and $|V^F| = |V| + 16|E| + 28\sum_{e\in E} k_e$. Then we deduce from Lemma 20:

$$\begin{aligned} \gamma(F) &= \tau(\widetilde{H}_q) + 4|E^{\widetilde{H}_q}| + |E| \\ &= \tau(G) + 5|E| + 9\sum_{e\in E} k_e \end{aligned} \tag{3.8}$$

We apply the hypothetical approximation algorithm `A` on F for different values of p, starting with $p = 2$ and augmenting it. While $p < \gamma(G'_q)$, the algorithm computes a solution C of probabilistic radius strictly greater than 2 (Lemma 18). Suppose now we set $p = \gamma(F) = \tau(G) + 5|E| + 9\sum_{e\in E} k_e$. Given Lemma 21, we obtain the following:

$$\mathbf{E}(C) = \frac{(|V^{\widetilde{H}_q}| - \tau(\widetilde{H}_q)) + 2(|V^F| - (|V^{\widetilde{H}_q}| - \tau(\widetilde{H}_q)))}{|V^F|} = \frac{2|V^F| - (|V^{\widetilde{H}_q}| - \tau(\widetilde{H}_q))}{|V^F|}$$

We deduce, using Relations 3.4, 3.7 and 3.8:

$$\begin{aligned} |V^F|\mathbf{E}(C) &= |V| + 32|E| + 55\sum_{e\in E} k_e + \tau(G) \\ &< 2|V| + 32|E| + 55\sum_{e\in E} k_e \end{aligned}$$

where the last inequality holds because $\tau(G) < |V|$. So, we have:

$$\mathbf{E}(C) < \frac{2|V| + 32|E| + 55 \sum\limits_{e\in E} k_e}{|V| + 16|E| + 28 \sum\limits_{e\in E} k_e} = 2 - \frac{\sum\limits_{e\in E} k_e}{|V| + 16|E| + 28 \sum\limits_{e\in E} k_e}$$

Using Equation 3.6, we have $\sum\limits_{e\in E} k_e \geq (q-1)|E|$. In addition, since G is of degree 2 or 3, we have $|V| \leq |E|$. It follows:

$$\mathbf{E}(C) < 2 - \frac{\sum\limits_{e\in E} k_e}{17|E| + 28 \sum\limits_{e\in E} k_e} \leq 2 - \frac{1}{28 + \frac{17|E|}{\sum\limits_{e\in E} k_e}} \leq 2 - \frac{1}{28 + \frac{17}{q-1}}$$

As $\varepsilon \geq \frac{17}{q-1}$ we get:

$$\mathbf{E}(C) \leq 2 - \frac{1}{28+\varepsilon} \leq \frac{55+2\varepsilon}{28+\varepsilon}$$

As a consequence, and since an optimal probabilistic solution C^* will satisfy $\mathbf{E}(C^*) \leq \mathbf{E}(C) \leq \frac{55+2\varepsilon}{28+\varepsilon}$, the approximation algorithm A will determine an approximated solution C in F of value:

$$\begin{array}{rcl} \mathbf{E}(C) & \leq & \rho \times \mathbf{E}(C^*) \\ & \leq & \rho \times \frac{55+2\varepsilon}{28+\varepsilon} \\ & < & \frac{56+2\varepsilon}{55+2\varepsilon} \times \frac{55+2\varepsilon}{28+\varepsilon} \\ & < & 2 \end{array} \tag{3.9}$$

Note that, given a solution C, computing its probabilistic radius can be done in polynomial time. Indeed, for any $v, s \in V^F$, computing $r^s(C, v)$ can be performed using any minimum path algorithm. Hence, we can apply successively the approximation algorithm A on the graph F for increasing values of p, starting with $p = 2$, until the computed solution C satisfies $\mathbf{E}(C) < 2$. Thanks to Lemma 18 and Equation 3.9, the algorithm stops for $p = \gamma(F) = \tau(G) + 5|E| + 9 \sum\limits_{e\in E} k_e$. Using Equations 3.8 we can deduce $\tau(G) = p - 5|E| - 9 \sum\limits_{e\in E} k_e$.

Since constructing $\widetilde{H}_q$ and F, as well as evaluating $\mathbf{E}(C)$, can be done in polynomial time, and since algorithm A will be run less than $|V|$ times, the whole process is polynomial. This is a contradiction if P≠NP, and the proof is complete. □

Conclusion

In this chapter, we outline the relation between `Min Vertex Cover` and `Min Dominating Set` with `RpCP` and `PpCP`. Based on these relations, we present different reductions establishing the hardness of `PpCP` and `RpCP`. We show that `RpCP` is NP-hard in all hereditary classes of uniform graphs where the decision version of `Min Vertex Cover` problem is NP-complete, and in all triangle-free classes of uniform graphs without pending vertices where the decision version of `Min Dominating Set` problem is NP-complete. We also prove that `RpCP` is NP-hard in subgrids. The main originality of our proof is the use of the intermediate graph $\widetilde{H}_q$ (see Figure 3.3): it can be seen as a perturbation of the subgrid H_q that leads to a hard class for `Min Vertex Cover`. Planar graphs and subgrids are relevant classes of graphs for our problem, as they are particularly suited for real case applications. Regarding `PpCP`, we propose two non-approximation results. The first result establishes that `PpCP` is not approximable with a ratio less than $\frac{20}{19}$ on bipartite planar graphs of degree 2 or 3. The second result states that `PpCP` is not approximable with a ratio less than $\frac{56}{55}$ on subgrids of degree at most 3. Even thought the latter result does not generalize the one we former obtained (the class is more restrictive but the new bound is closer to 1), the proof requires a much deeper analysis.

In the next chapter, we propose polynomial methods to generate exact and approximation results for `PpCP`.

CHAPTER 4

POLYNOMIAL AND APPROXIMATION RESULTS FOR PpCP

Abstract

In this chapter we present an explicit solution for `PpCP` on uniform paths. The proof requires a decomposition of `PpCP` in multiple auxiliary subproblems. In addition, we propose some approximation results for `PpCP` on edge-weighted graphs with bounded edge lengths. These results require approximation results for two variants of `Min p-Center` called `Min MAC p-Center` and `Min Partial p-Center`.

Contents

4.1 An explicit solution for PpCP on paths in uniform case 70

4.1.1 Additional notations and preliminary results 70

4.1.2 Auxiliary sub-problems 73

4.1.3 An optimal solution 76

4.1.4 The case of edge-weighted path 81

4.2 Approximation algorithms 83

4.2.1 A polynomial approximation preserving reduction . . . 84

4.2.2 Constant approximation algorithms 87

Introduction

We have seen in the previous chapter that PpCP is NP-hard on planar graphs and subgrids of degrees $\{2, 3\}$. In this chapter, we study polynomial methods to compute solutions for PpCP with a guarantee of performance.

In Section 4.1, we describe an explicit optimal solution for PpCP on paths in the uniform case. The results of this section have been published as a conference paper in [27]. In Section 4.2, we show that PpCP admits a polynomial approximation algorithm for PpCP guaranteeing the ratio 3 on trees in the uniform case. In addition, for edge-weighted graphs with bounded edge lengths, we present polynomial approximation algorithm for PpCP guaranteeing ratio $4\overline{deg}(G) + 2$, where $\overline{deg}(G)$ is the average degree of G. The results of this section have been submitted in [78].

4.1 An explicit solution for PpCP on paths in uniform case

The main result of this section is the explicit optimality of a *monotone balanced p-set* (see Definitions 23, 29) for PpCP on paths in the uniform case. However, the proof is non-trivial and require a decomposition of the problem.

First, in Subsection 4.1.1 we will present some additional notations and definitions specific for PpCP on paths as well as first preliminary results. In Subsection 4.1.2, we introduce different subproblems of PpCP which allow us to decompose our problem. In Subsection 4.1.2, we prove that a monotone balanced solution is optimum for PpCP on paths. In Subsection 4.1.4, we discuss the case of edge-weighted paths.

4.1.1 Additional notations and preliminary results

We denote $P_n = \{1, \ldots, n\}$ a path on n vertices. For a given $p \geq p' \geq 2$, consider a p-set $C = \{c_1, \ldots, c_{p'}\} \in \mathcal{C}_p(P_n)$ where $\mathcal{C}_p(P_n)$ is the set of feasible solutions for PpCP as defined in Section 2.1.5. In this section, we consider $p' = p$. We will see shortly after that an optimal solution for PpCP on a path is always of size p. Then C induces $p - 1$ *segments* μ_i^C of length λ_i^C, $i = 1, \ldots, p - 1$, where a segment corresponds to the path between two consecutive centers. The vertices between the two centers are called *internal* vertices. Conversely, we observe that any vector $\lambda_1, \ldots, \lambda_{p-1} \in \mathbb{N}^{*p-1}$ with $\sum_{i=1}^{p-1} \lambda_i = n - 1$ corresponds to an unique feasible solution on P_n. Thus we can consider equivalently a p-set $C \subseteq V$ of size p or the corresponding vector $\lambda(C) = (\lambda_1^C, \ldots, \lambda_{p-1}^C)$. In the following, to harmonize notations, for any p-set C we define $\mu_0^C = \{1\}, \mu_p^C = \{n\}$ and $\lambda_0^C = \lambda_p^C = 0$.

Example. In Figure 4.1, $C = \{1, 6, 9, 17\}$ is a feasible solution of PpCP for $p = 4$ on P_{17}. C induces 3 segments μ_1^C, μ_2^C and μ_3^C and $\lambda(C) = \{5, 3, 8\}$.

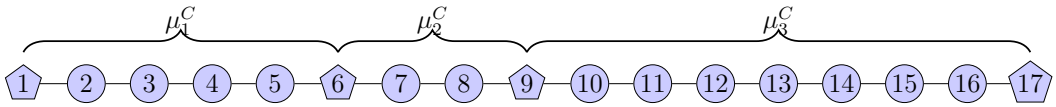


Figure 4.1: Segments induced by solution $\{1, 6, 9, 17\}$ on P_{17}

♣

Two specific kinds of solutions emerge on paths, which we define next.

Definition 23. On a path, a p-set $C \in \mathcal{C}_p(P_n)$ with $\lambda(C) = (\lambda_1^C, \ldots, \lambda_{p-1}^C)$ is called *balanced* if $\max_{i,j \in \{1,\ldots,p-1\}} |\lambda_i^C - \lambda_j^C|$ is minimum over $\mathcal{C}_p(P_n)$.

We directly outline a characteristic of balanced p-sets on uniform paths.

Corollary 24. On a path in the uniform case, for any $i \in \{1, \ldots, p-1\}$, a balanced p-set C verifies $\lambda_i^C \in \{\lfloor \frac{n-1}{p-1} \rfloor, \lceil \frac{n-1}{p-1} \rceil\}$

Finally we propose a system of equations to compute the evacuation radius of any feasible solution for any scenario.

Proposition 25. On P_n, the radius of $C \in \mathcal{C}_p(P_n)$ for scenario s is given by:

- if $s \in C$ and $i \in \{1, \ldots, p-1\}$ such that $s = c_i$:

$$r^s(C) = \max\{\lambda_{i-1}^C - 1, \lambda_i^C - 1, \max_{q \in \{1,\ldots,p-1\}; s \notin \mu_q^C} \lfloor \frac{\lambda_q^C}{2} \rfloor\} \tag{4.1}$$

- if $s \notin C$ and $i \in \{1, \ldots, p-1\}$ such that $s \in \mu_i^C$:

$$r^s(C) = \max\{s - c_i, c_{i+1} - s, \max_{q \in \{1,\ldots,p-1\}; s \notin \mu_q^C} \lfloor \frac{\lambda_q^C}{2} \rfloor\} \tag{4.2}$$

Proof. For scenario s, let Z_s be the set of segments including vertex s, i.e. $Z_s = \{\mu_j^C : s \in \mu_j^C, j \in \{1, \ldots, p-1\}\}$. If $s \notin C$, then $|Z_s| = 1$, else $|Z_s| = 2$. We can write $r^s(C) = \max\{\max_{v \in Z_s} r^s(C, v), \max_{v \in P_n \setminus Z_s} r^s(C, v)\}$.

We can see that, for any vertex $v \in P_n \setminus Z_s$, its evacuation distance in scenario s is equal to its distance to C in G^s, which is the same than its distance to C in G i.e., $r^s(C, v) = d^s(v, C) = d(v, C)$. We obtain:

$$r^s(C) = \max\{\max_{v \in Z_s} r^s(C, v), \max_{v \in P_n \setminus Z_s} d(v, C)\} \tag{4.3}$$

We analyze first $\max_{v \in P_n \setminus Z_s} d(v, C)$. Let $q \in \{1, \ldots, p-1\}$ such that $\mu_q^C \notin Z_s$. In scenario s, we observe that the evacuation distance of any vertex on μ_q^C is less than or equal to the evacuation distance of the middle vertex (or vertices) of μ_q^C. The evacuation distance of the middle vertex (or vertices) is $\lfloor \frac{\lambda_q^C}{2} \rfloor$, thus we have:

$$\max_{v \in P_n \setminus Z_s} d(v, C) = \max_{q \in \{1,\ldots,p-1\} : s \notin \mu_q^C} \lfloor \frac{\lambda_q^C}{2} \rfloor \tag{4.4}$$

Next we analyze $\max_{v \in Z_s} r^s(C, v)$. Two cases emerge:

1. $s \in C$: we define i such that $c_i = s$, and μ_{i-1}^C, μ_i^C are the two segments adjacent to s. In scenario s, $r^s(C, s) = 0$ and the evacuation paths of the internal vertices of segments μ_{i-1}^C and μ_i^C are included in the evacuation paths of vertices $(s-1)$ and $(s+1)$ respectively. Thus we have: $\max_{v \in Z_s} r^s(C, v) = \max\{r^s(C, s-1), r^s(C, s+1)\} = \max\{d^s(s-1, C), d^s(s+1, C)\} = \max\{\lambda_i^C - 1, \lambda_{i+1}^C - 1\}$. Combined with Equations 4.3,4.4, we obtain Equation 4.1.

2. $s \notin C$: we define i such that $c_i < s < c_{i+1}$. Then we have $r^s(C, s) = 1 + \max\{d^s(s-1, C), d^s(s+1, C)\} = 1 + \max\{s - 1 - c_i, c_{i+1} - (s+1)\} = \max\{s - c_i, c_{i+1} - s\}$. In addition, we observe that all the evacuation paths of the vertices of $\mu_i^C - s$ are included in the evacuation path of $s-1$ or $s+1$. Thus we have: $\max_{v \in Z_s} r^s(C, v) = \max\{s - c_i, c_{i+1} - s\}$ Combined with Equations 4.3,4.4, we obtain Equation 4.2.

□

Remark 3. Given Proposition 25, note that adding shelters can only contribute to decrease the evacuation radius of the solution. For this reason, in this section, we consider that only p-sets of maximum size.

Example. Illustrating our proposition, we will give examples based on Figure 4.1. By Equation 4.2, the disrupted radius of C for scenario 3 is :

$$\begin{aligned} r^3(C) &= \max\{3 - c_1, c_2 - 3, \max\{\lfloor \tfrac{\lambda_2^C}{2} \rfloor, \lfloor \tfrac{\lambda_3^C}{2} \rfloor\}\} \\ &= \max\{3 - 1, 6 - 3, \max\{\lfloor \tfrac{3}{2} \rfloor, \lfloor \tfrac{8}{2} \rfloor\}\} \\ &= \max\{2, 3, \max\{1, 4\}\} \\ &= 4 \end{aligned}$$

We observe that, although the fire happens on segment μ_1^C, the radius of the solution C for scenario 3 is induced by the evacuation distance of people on μ_3^C.

For scenario 9, the disrupted radius is, by Equation 4.1:

$$\begin{aligned} r^9(C) &= \max\{\lambda_2^C - 1, \lambda_3^C - 1, \max\{\lfloor \tfrac{\lambda_1^C}{2} \rfloor\}\} \\ &= \max\{3-1, 8-1, \lfloor \tfrac{5}{2} \rfloor\} \\ &= \max\{2, 7, 2\} \\ &= 7 \end{aligned}$$

♣

We can prove already, using Proposition 25, that a balanced solution is optimal for PpCP on P_n when $\frac{n}{2} < p < n$.

Proposition 26. On paths, for $\frac{n}{2} < p < n$, a balanced solution is optimum for PpCP.

Proof. We prove that any feasible solution has a probabilistic radius of value higher than or equal to the probabilistic radius of a balanced solution.

For P_n, denote B a balanced solution and λ^B its segment vector. By Corollary 24, we know that $\lambda_q^B \in \{\lfloor \frac{n-1}{p-1} \rfloor, \lceil \frac{n-1}{p-1} \rceil\}, \forall q \in \{1, \ldots, p-1\}$. On the first hand, as $p < n$, we have $\frac{n-1}{p-1} > 1$. On the other hand, as $p > \frac{n}{2}$ and p is integer, we have $p \geq \frac{n+1}{2}$ which induces $1 < \frac{n-1}{p-1} \leq 2$. It follows that, $\forall i \in \{1, \ldots, p-1\}$ we have $\lambda_i^B \leq 2$, and there exists at least $j \in \{1, \ldots, p-1\}$ such that $\lambda_j^B = 2$. Hence, by Proposition 25 we have: $r^s(B) = 1, \forall s \in V$.

Now suppose C is a non-balanced solution on P_n. Then there exists at least $i \in \{1, \ldots, p-1\}$ such that $\lambda_i^C \geq 3$. Based on Proposition 25, it follows that

- for $s \in P_n \setminus \mu_i^C$, we get $r^s(C) \geq \lfloor \frac{\lambda_i^C}{2} \rfloor \geq 1$.
- for $s \in \mu_i^C$, we get $r^s(C) \geq 2$.

Consequently, $\mathbf{E}(C) = \sum_{s \in V} \frac{1}{n} r^s(C) > \sum_{s \in V} \frac{1}{n} r^s(B) = \mathbf{E}(B)$. Thus, any feasible solution has a probabilistic radius of value higher than or equal to the probabilistic radius of a balanced solution and the proof is complete.

□

The case $2 \leq p \leq \frac{n}{2}$ is more complex and requires a decomposition of the problem into subproblems. We introduce next the auxiliary subproblems we will use to identify an exact solution for PpCP on path when $2 \leq p \leq \frac{n}{2}$.

4.1.2 Auxiliary sub-problems

In this section, we introduce some subproblems of PpCP, where the evacuation distance of a restricted number of vertices is considered rather than all the vertices of the graph.

4.1.2.1 The local PpCP problem

We call *local area* of a vertex v the close neighborhood $(\Gamma(s) \cup \{s\})$ of v. The *local evacuation radius* of a solution C for a scenario s is the maximum evacuation distance of a vertex in the close neighborhood of s. It is given by:

$$\widehat{r}^s(C) = \max_{v \in \{s\} \cup \Gamma(s)} r^s(C, v) \tag{4.5}$$

Remark 4. For $s \notin C$, $\widehat{r}^s(C) = r^s(C, s)$. If $s \in C$, then $\widehat{r}^s(C) = \max_{v \in \Gamma(s)} r^s(C, v))$.

Accordingly, we define the objective function $\widehat{\mathbf{E}}$, called local probabilistic radius, such that:

$$\widehat{\mathbf{E}}(C) = \sum_{s \in V} \frac{1}{n} \widehat{r}^s(C)$$

On G and for a given p, the local PpCP problem is then the problem of finding $C \in \mathcal{C}_p(P_n)$ that minimizes $\widehat{\mathbf{E}}(C)$.

Note that, $\widehat{r}^s(C) \leq r^s(C)$ and $\widehat{\mathbf{E}}(C) \leq \mathbf{E}(C)$. In addition, from Proposition 25 we can induce the following corollary:

Corollary 27. On P_n, consider $C \in \mathcal{C}_p(P_n)$, we have:

- if $s \in C$ and $i \in \{1, \ldots, p-1\}$ such that $s \in \mu_{i-1}^C \cap \mu_i^C$:

$$\widehat{r}^s(C) \ = \max\{\lambda_{i-1}^C - 1, \lambda_i^C - 1\}$$

- if $s \notin C$ and $i \in \{1, \ldots, p-1\}$ such that $s \in \mu_i^C$:

$$\widehat{r}^s(C) \ = \max\{s - c_i, c_{i+1} - s\}$$

4.1.2.2 PpCP restricted to a subset of scenarios

As we consider an uniform probability distribution, the value of $\mathbf{E}(C)$ is proportional to the sum of the evacuation radius of C over all scenarios. We propose then the following decomposition of $\mathbf{E}(C)$:

$$\mathbf{E}(C) \ = \mathbf{E}_C(C) + \mathbf{E}_{\overline{C}}(C)$$

where

$$\begin{aligned} \mathbf{E}_C(C) &= \sum_{s \in C} \tfrac{1}{n} r^s(C) \\ \mathbf{E}_{\overline{C}}(C) &= \sum_{s \in V \setminus C} \tfrac{1}{n} r^s(C) \end{aligned}$$

In other words, $\mathbf{E}_C(C)$ is the contribution to $\mathbf{E}(C)$ of scenarios for which the fire occurs in C, while $\mathbf{E}_{\bar{C}}(C)$ is the contribution to $\mathbf{E}(C)$ of scenarios for which the fire occurs on another vertex. Note that, if a solution is optimal for both components, then it is optimal for the whole problem. We can treat these components as two different problems. In addition, we define similarly $\widehat{\mathbf{E}}_C(C)$ and $\widehat{\mathbf{E}}_{\overline{C}}(C)$ such that:

$$\widehat{\mathbf{E}}(C) = \widehat{\mathbf{E}}_C(C) + \widehat{\mathbf{E}}_{\overline{C}}(C)$$

Using the previous decomposition, for $2 \leq p \leq \frac{n}{2}$, we prove a structural property of a balanced solution.

Lemma 28. On P_n, for $2 \leq p \leq \frac{n}{2}$, a balanced solution B satisfies

$$\mathbf{E}_C(B) = \widehat{\mathbf{E}}_C(B), \mathbf{E}_{\bar{C}}(B) = \widehat{\mathbf{E}}_{\bar{C}}(B)$$

and thus

$$\mathbf{E}(B) = \widehat{\mathbf{E}}(B)$$

Proof. For $2 \leq p \leq \frac{n}{2}$, consider a balanced solution B on P_n.

- First we prove $E_C(B) = \widehat{E}_C(B)$. As $p \leq \frac{n}{2}$, then $\frac{n-1}{p-1} > 2$ and $\left\lfloor \frac{n-1}{p-1} \right\rfloor \geq 2$. We obtain:

  $$\max_{q \in \{1,\dots,p-1\}} \lfloor \frac{\lambda_q^B}{2} \rfloor \leq \left\lfloor \frac{1}{2} \left\lceil \frac{n-1}{p-1} \right\rceil \right\rfloor \leq \left\lfloor \frac{n-1}{p-1} \right\rfloor - 1$$

  By Corollary 24, we know that $\lambda_q^B \in \{\lfloor \frac{n-1}{p-1} \rfloor, \lceil \frac{n-1}{p-1} \rceil\}, \forall q \in \{1,\dots,p-1\}$. Given previous equation, for any given $i \in \{1,\dots,p-1\}$ we then have $\max_{q \in \{1,\dots,p-1\}} \lfloor \frac{\lambda_q^B}{2} \rfloor \leq \lambda_i^B - 1$. Given Proposition 25 and Corollary 27, we can deduce $r^s(B) = \hat{r}^s(B), \forall s \in C$. Thus $E_C(B) = \widehat{E}_C(B)$.

- Second we prove $E_{\bar{C}}(B) = \widehat{E}_{\bar{C}}(B)$. In line with Proposition 25, we observe that, $\forall i \in \{1,\dots,p-1\}$ and $\forall s \in \mu_i^B \setminus C$, there exists at least one vertex in $\mu_i^B \setminus C$ whose evacuation path includes at least one half of segment μ_i^B. Therefore:

  $$r^s(B) \geq \left\lceil \frac{\lambda_i^B}{2} \right\rceil \geq \left\lceil \frac{1}{2} \left\lfloor \frac{n-1}{p-1} \right\rfloor \right\rceil \geq \left\lfloor \frac{1}{2} \left\lceil \frac{n-1}{p-1} \right\rceil \right\rfloor \geq \max_{q \in \{1,\dots,p-1\}} \lfloor \frac{\lambda_q^B}{2} \rfloor$$

  Given Proposition 25 and Corollary 27, we can deduce $r^s(B) = \hat{r}^s(B) : \forall s \in V \setminus C$. Thus $E_{\bar{C}}(B) = \widehat{E}_{\bar{C}}(B)$.

From the previous results we get: $E(B) = E_C(B) + E_{\bar{C}}(B) = \widehat{E}_{\bar{C}}(B) + \widehat{E}_C(B) = \widehat{E}(B)$. □

Lemma 28 induces a method for us to find an optimal solution for `PpCP`: if we can find a balanced solution minimizing $\widehat{E}$, then this solution minimizes E. In the next section we prove such solution exists, and we provide it.

4.1.3 An optimal solution

In this section, we give an optimal solution for PpCP on an uniform path. To this end, we need one last definition.

Definition 29. A solution $C \in \mathcal{C}_p(P_n)$ with $\lambda(C) = (\lambda_1^C, \ldots, \lambda_{p-1}^C)$ is called *monotone* if $\forall i, j \in \{1, \ldots, p-1\}, i < j$ we have $\lambda_i^C \leq \lambda_j^C$ (or inversely $\lambda_i^C \geq \lambda_j^C$).

In the following, we consider only monotone non-decreasing solutions. It is straightforward to prove the same results on monotone non-increasing solutions.

Before proceeding, we show an example of an instance where a monotone balanced solution is optimal for PpCP while another balanced solution is not optimal for PpCP.

Example. Consider PpCP with $p = 4$ on P_{12}. In the following Figures 4.2 and 4.3 , for a given solution $\{c_1, c_2, c_3, c_4\}$ we display under each center c_i the evacuation radius of the solution for scenario c_i. In Figure 4.2 we have a non-monotone balanced solution $C_1 = \{1, 5, 8, 12\}$ inducing the segments μ_1^C, μ_2^C and μ_3^C of length 4 , 3 and 4 respectively. C_1 is non-monotone as $\lambda_2 < \lambda_1 < \lambda_3$ and $\mathbf{E}_C(C_1) = \frac{3+3+3+3}{12} = 1$.

Meanwhile, in Figure 4.3 we have a monotone solution $C_2 = \{1, 4, 8, 12\}$ inducing the segments μ_1^C, μ_2^C and μ_3^C of length 3 , 4 and 4 respectively. C_2 is monotone as $\lambda_1 \leq \lambda_2 \leq \lambda_3$ and $\mathbf{E}_C(C_2) = \frac{2+3+3+3}{12} = \frac{11}{12}$.

Thus, $\mathbf{E}_C(C_2) < \mathbf{E}_C(C_1)$. In Lemma 34, we will see that $\mathbf{E}_{\bar{C}}(C_1) = \mathbf{E}_{\bar{C}}(C_2)$ as C_1 and C_2 induce segments with the same lengths. Consequently $\mathbf{E}(C_2) < \mathbf{E}(C_1)$.

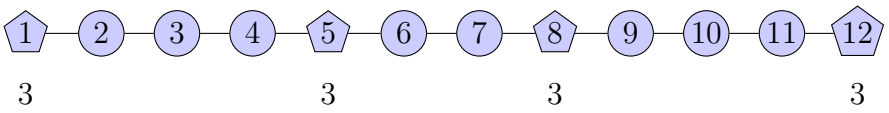


Figure 4.2: Evacuation radius induced by scenarios $s \in C$ where $C = \{1, 5, 8, 12\}$ on uniform P_{12}.

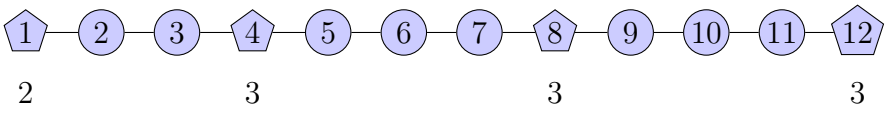


Figure 4.3: Evacuation radius induced by scenarios $s \in C$ where $C = \{1, 4, 8, 12\}$ on uniform P_{12}.

♣

As we have already proved that a balanced solution is optimal when $\frac{n}{2} < p < n$ (Proposition 26), we now focus on the case $2 < p \leq \frac{n}{2}$. In Section 4.1.3.1, we prove that any monotone balanced solution minimizes $\mathbf{E}_C(C), \forall C \in \mathcal{C}_p(P_n)$. In Section 4.1.3.2, we prove that any monotone balanced solution minimizes $\mathbf{E}_{\bar{C}}(C), \forall C \in \mathcal{C}_p(P_n)$. Thus, by Lemma 28 a monotone balanced solution minimizes $\mathbf{E}(C), \forall C \in \mathcal{C}_p(P_n)$, and we can state:

Theorem 30. On P_n, for $p > 2$, a monotone balanced solution is optimal for `PpCP`.

4.1.3.1 The contribution of the centers

In this subsection, we show that a monotone balanced solution minimizes $\widehat{\mathbf{E}}_C(C)$ and consequently minimizes $\mathbf{E}_C(C)$ too. Note that, based on Proposition 25, for a given solution C we have:

$$\widehat{\mathbf{E}}_C(C) \;\; = \tfrac{1}{n} \cdot \left(\lambda_1^C + \sum_{i=2}^{p-1} \max(\lambda_{i-1}^C, \lambda_i^C) + \lambda_{p-1}^C \right) \tag{4.6}$$

In the following, we define $\lambda_{\leq} = (\lambda_{i_1}, \ldots, \lambda_{i_\kappa})$, where $i : \{1, \ldots, p-1\} \to \{1, \ldots, p-1\}$ is a permutation such that $\lambda_{i_1} \leq \ldots \leq \lambda_{i_{p-1}}$, a monotone solution induced by λ. For given solution C, we denote $C_{\leq}$ the solution such that $\lambda(C_{\leq}) = \lambda_{\leq}(C)$.

Lemma 31. For any $C \in \mathcal{C}_p(P_n)$ we have: $\widehat{\mathbf{E}}_C(C) \geq \widehat{\mathbf{E}}_C(C_{\leq})$

Proof. Let $C \in \mathcal{C}_p(P_n)$, we denote $\lambda(C) = (\lambda_1, \ldots, \lambda_{p-1})$. If $\lambda_1 = \max\{\lambda_i, i = 1, \ldots p-1\}$, then we define $r = 0$, else let r be the maximum index in $\{1, \ldots, p-1\}$ such that $\lambda_1 \leq \ldots \leq \lambda_r$ and $\forall j \geq r, \lambda_j \geq \lambda_r$.

If $r = p-1$, then C is monotone, thus $C = C_{\leq}$ and the lemma is verified. The value $r = p-2$ is not possible: if $r = p-2$ then $\lambda_1 \leq \ldots \leq \lambda_{p-2}$ and $\lambda_{p-2} \leq \lambda_{p-1}$ which is actually the case $r = p-1$. Suppose $r < p-2$. We consider $t \in \text{argmin}\ \{\lambda_j, j = r+1, \ldots, p-1\}$: by definition, we have $\lambda_{r+1} \geq \lambda_r$ if $r > 0$, $t > r+1$ and $\lambda_r \leq \lambda_t < \lambda_{r+1}$. Let λ' be the vector obtained from λ by moving the t^{th} coordinate at the position $r+1$:

$$\begin{array}{lcll} \lambda'_i & = & \lambda_i & , i = 1, \ldots, r \\ \lambda'_{r+1} & = & \lambda_t & \\ \lambda'_i & = & \lambda_{i-1} & , i = r+2, \ldots, t \\ \lambda'_i & = & \lambda_i & , i > t \end{array}$$

We claim that $\widehat{E}_C(\lambda) \geq \widehat{E}_C(\lambda')$. Indeed, suppose first $t < p-1$ and consider the expression of $\widehat{E}_C(\lambda')$ and $\widehat{E}_C(\lambda)$, as sums of p terms (see Equation 4.6). The three terms $\max(\lambda_r, \lambda_{r+1}), \max(\lambda_{t-1}, \lambda_t)$ and $\max(\lambda_t, \lambda_{t+1})$ in the expression of $\widehat{E}_C(\lambda)$

are replaced respectively in the expression of $\widehat{E}_C(\lambda')$ by $\max(\lambda_r, \lambda_t), \max(\lambda_t, \lambda_{r+1})$ and $\max(\lambda_{t-1}, \lambda_{t+1})$ and all the other terms are identical in both expressions. We conclude:

$$\begin{array}{rcl} \widehat{E}_C(\lambda) - \widehat{E}_C(\lambda') = & \frac{1}{n}\left(\max(\lambda_r, \lambda_{r+1}) + \max(\lambda_{t-1}, \lambda_t) + \max(\lambda_t, \lambda_{t+1})\right. \\ & \left. - [\max(\lambda_r, \lambda_t) + \max(\lambda_t, \lambda_{r+1}) + \max(\lambda_{t-1}, \lambda_{t+1})]\right) \\ = & \frac{1}{n}\left(\lambda_{r+1} + \lambda_{t-1} + \lambda_{t+1} - \lambda_t - \lambda_{r+1} - \max(\lambda_{t-1}, \lambda_{t+1}))\right) \\ = & \frac{1}{n}\left(\lambda_{t-1} + \lambda_{t+1} - \lambda_t - \max(\lambda_{t-1}, \lambda_{t+1})\right) \\ \geq & 0 \end{array}$$

where the last inequality holds since $\lambda_t \leq \min(\lambda_{t-1}, \lambda_{t+1})$.

Suppose now $t = p - 1$, then a similar approach without the terms involving $t + 1$ leads to:

$$\begin{array}{rcll} \widehat{E}_C(\lambda) - \widehat{E}_C(\lambda') = & \frac{1}{n} & (\max(\lambda_r, \lambda_{r+1}) + \max(\lambda_{p-1-1}, \lambda_{p-1}) - \\ & & [\max(\lambda_r, \lambda_{p-1}) + \max(\lambda_{p-1}, \lambda_{r+1})]) \\ = & \frac{1}{n} & (\lambda_{r+1} + \lambda_{p-1-1} - \lambda_{p-1} - \lambda_{r+1}) \\ \geq & 0 & \end{array}$$

The last inequality is valid since $\lambda_{p-1-1} \geq \lambda_{p-1}$. Note that these arguments hold also if $r = 0$. The proposition is deduced by induction. □

We are now ready to show the main result of this subsection:

Proposition 32. $B \in \operatorname{argmin} \{\mathbf{E}_C(C), C \in \mathcal{C}_p(P_n)\}$.

Proof. First we claim that $B \in \operatorname{argmin} \{\widehat{\mathbf{E}}_C(C), C \in \mathcal{C}_p(P_n)\}$. Indeed in order to minimize $\widehat{\mathbf{E}}_C(C)$, Lemma 31 ensures we can restrict ourselves to monotone solutions. But for any monotone C, we have $\widehat{\mathbf{E}}_C(C) = \frac{1}{n}(n + \lambda^C_{p-1})$ and consequently a monotone solution minimizing $\widehat{\mathbf{E}}_C(C)$ is obtained by solving:

$$\begin{cases} \min & \lambda_{p-1} \\ & \lambda_1 \leq \ldots \leq \lambda_{p-1} \\ & \sum_{i=1}^{p-1} \lambda_i = n, \ \lambda \in \mathbb{N}^{p-1} \end{cases}$$

This system admits B as an optimal solution. Therefore, by Lemma 28 we have: $\mathbf{E}_C(B) = \widehat{\mathbf{E}}_C(B) = \min \widehat{\mathbf{E}}_C(C) \leq \min \mathbf{E}_C(C), C \in \mathcal{C}_p(P_n)$ and consequently $B \in \operatorname{argmin} \{\mathbf{E}_C(C), C \in \mathcal{C}_p(P_n)\}$.

□

4.1.3.2 The contribution of the internal vertices

In this subsection, we show that a monotone balanced solution minimizes $\mathbf{E}_{\bar{C}}(C)$, $\forall C \in \mathcal{C}_p(P_n)$.

Lemma 33. Given an initial solution $C \in \mathcal{C}_p(P_n)$, let $1 \leq b < a \leq p-1$ and $1 \leq m \leq n-1$ such that $\lambda_a^C + \lambda_b^C = m, \lambda_a^C \geq \lambda_b^C$ and $|\lambda_a^C - \lambda_b^C| \geq 2$. There is a solution $C' \in \mathcal{C}_p(P_n)$ such that $\lambda_a^{C'} + \lambda_b^{C'} = m, |\lambda_a^{C'} - \lambda_b^{C'}| \leq 1$, $\lambda_a^{C'} \geq \lambda_b^{C'}$ and $\forall i \in \{1, \ldots, p-1\}, i \neq a, b, \lambda_i^{C'} = \lambda_i$ which verifies:

$$\widehat{\mathbf{E}}_{\bar{C}}(C') \leq \widehat{\mathbf{E}}_{\bar{C}}(C)$$

Proof. Let $C = \{c_1, \ldots, c_{p-1}\}$, we denote $\alpha(\lambda_a^C) = \sum_{v \in \mu_a^C \setminus C} \max\{v - c_a, c_{a+1} - v\}$ the contribution of μ_a^C to the value of $\widehat{E}_{\bar{C}}(C)$. Then $\widehat{E}_{\bar{C}}(C) = \frac{1}{n} \sum_{i=1,\ldots,p-1} \alpha(\lambda_i^C)$. We denote also A a logical proposition, and by $\mathbb{1}_A$ the boolean function such that $\mathbb{1}_A = 1$ if A is true, and $\mathbb{1}_A = 0$ if A is false. Note that $c_{a+1} - c_a = \lambda_a^C$. Then, for $v \in \lambda_a^C$, we have:

$$\begin{aligned}
\alpha(\lambda_a^C) &= \sum_{v \in \mu_a^C \setminus C} \max\{v - c_a, c_{a+1} - v\} \\
&= \sum_{i=1}^{\lambda_a^C - 1} \max\{i, \lambda_a^C - i\} \\
&= 2 \cdot \sum_{j=\lceil \frac{\lambda_a^C}{2} \rceil}^{\lambda_a^C - 1} (j) - \mathbb{1}_{(\lambda_a^C \text{ even})}(\tfrac{\lambda_a^C}{2}) \\
&= \lfloor \tfrac{\lambda_a^C}{2} \rfloor (\lambda_a^C + \lceil \tfrac{\lambda_a^C}{2} \rceil - 1) - \mathbb{1}_{(\lambda_a^C \text{ even})}(\tfrac{\lambda_a^C}{2})
\end{aligned}$$

The previous result applies also to $\alpha(\lambda_b^C)$. As $\lambda_b^C = m - \lambda_a^C$, we can express $\alpha(\lambda_a^C) + \alpha(\lambda_b^C)$ as a function of λ_a^C:

$$\begin{aligned}
\alpha(\lambda_a^C) + \alpha(\lambda_b^C) = \; & \lfloor \tfrac{\lambda_a^C}{2} \rfloor (\lambda_a^C + \lceil \tfrac{\lambda_a^C}{2} \rceil - 1) - \mathbb{1}_{(\lambda_a^C \text{ even})}(\tfrac{\lambda_a^C}{2}) \\
+ \; & \lfloor \tfrac{m - \lambda_a^C}{2} \rfloor (m - \lambda_a^C + \lceil \tfrac{m - \lambda_a^C}{2} \rceil - 1) - \mathbb{1}_{((m - \lambda_a^C) \text{ even})}(\tfrac{m - \lambda_a^C}{2})
\end{aligned}$$

If we study the different combinations of parities of λ_a^C and m, we get:

$$\tfrac{3}{2}(\lambda_a^C)^2 - \tfrac{3n}{2}\lambda_a^C + \tfrac{3}{4}m^2 - m \leq \alpha(\lambda_a^C) + \alpha(\lambda_b^C) \leq \tfrac{3}{2}(\lambda_a^C)^2 - \tfrac{3m}{2}\lambda_a^C + \tfrac{3}{4}m^2 - m + \tfrac{1}{2}$$

Defining the function $f(\lambda_a^C) = \frac{3}{2}(\lambda_a^C)^2 - \frac{3m}{2}\lambda_a^C$, we get:

$$\alpha(\lambda_a^C) + \alpha(\lambda_b^C) - \frac{1}{2} \leq f(\lambda_a^C) + \frac{3}{4}m^2 - m \leq \alpha(\lambda_a^C) + \alpha(\lambda_b^C) \tag{4.7}$$

The same holds for $(\lambda_a^{C'}, \lambda_b^{C'})$. In Equation 4.7, note that $\frac{3}{4}m^2 - m$ is constant and that $f(\lambda_a^C)$ is a monotone function for $\lambda_a^C > \frac{m}{2}$. Therefore, since $\lambda_a^{C'} \leq \lambda_a^C$, we have $f(\lambda_a^{C'}) \leq f(\lambda_a^C)$. We deduce from Equation 4.7: $\alpha(\lambda_a^{C'}) + \alpha(\lambda_b^{C'}) \leq \alpha(\lambda_a) + \alpha(\lambda_b)$.

Finally, as $\alpha(\lambda_i^C) = \alpha(\lambda_i^{C'}), \forall i \in \{1, \ldots, k-1\}, i \neq a, b$, we conclude $\widehat{E}_{\bar{C}}(C') \leq \widehat{E}_{\bar{C}}(C)$. □

Lemma 34. For any $B, C \in \mathcal{C}_p(P_n)$ such that B is monotone and balanced, we have:

$$\widehat{\mathbf{E}}_{\bar{C}}(C) \geq \widehat{\mathbf{E}}_{\bar{C}}(B)$$

Proof. Let $C \in \mathcal{C}_p(P_n)$. We observe that for any internal vertex $s \in V \setminus C$, $\widehat{r}^s(C)$ depends only on the position of s in its segment. It follows that any permutation on the segments does not impact the overall sum of the local evacuation radius, which induces: $\widehat{\mathbf{E}}_{\bar{C}}(C) = \widehat{\mathbf{E}}_{\bar{C}}(C_{\leq})$.

Therefore, we suppose in the following, without loss of generality, that $\lambda(C)$ is monotone. We look at the pair of segments $(\lambda_1^C, \lambda_{p-1}^C)$ having the largest length difference. If $|\lambda_1^C - \lambda_{p-1}^C| \leq 1$, then C is a monotone balanced solution and the lemma is verified. Otherwise, we create a new solution C' by replacing the extreme segments by a new pair of segments of size $\lambda_1^{C'} = \lfloor \frac{n - \sum_{i=2}^{p-1-1} \lambda_i^C}{2} \rfloor$ and $\lambda_{p-1}^{C'} = \lceil \frac{n - \sum_{i=2}^{p-1-1} \lambda_i^C}{2} \rceil$. As $\lambda_1^C + \lambda_{p-1}^C = \lambda_1^{C'} + \lambda_{p-1}^{C'}$, we deduce by Lemma 33 that $\widehat{\mathbf{E}}_{\bar{C}}(C) \geq \widehat{\mathbf{E}}_{\bar{C}}(C')$.

Let C'' be the monotone solution induced by $\lambda^{C'}$ i.e., $\lambda(C'') = \lambda_{\leq}^{C'}$. Then, $\widehat{\mathbf{E}}_{\bar{C}}(C') = \widehat{\mathbf{E}}_{\bar{C}}(C'')$ and $\forall j \in \{1, \ldots, p-1\}, \lambda_1^C \leq \lambda_j^{C''} \leq \lambda_{p-1}^C$. We can make two observations:

- $\lambda_1^{C''} \geq \lambda_1^C$ and $\lambda_{p-1}^{C''} \leq \lambda_{p-1}^C$, thus $\lambda_{p-1}^{C''} - \lambda_1^{C''} \leq \lambda_{p-1}^C - \lambda_1^C$.

  It means that the maximum length difference between segments in the newly created solution doesn't increase compared to the original solution.

- $\lambda_1^{C'} > \lambda_1^C$ and $\lambda_{p-1}^{C'} < \lambda_{p-1}^C$.

  This ensures that after at least $\frac{n}{2}$ iterations, the maximum length difference strictly decreases.

Therefore we can iterate this process with the new extreme segments ($\lambda_1^{C''}$ and $\lambda_{p-1}^{C''}$) until we get a solution whose extreme segments lengths differ by at most 1, in which case all segments differ by at most 1. This is then a balanced solution, hereby the proof is completed.

□

Proposition 35.

$$B \in \operatorname{argmin} \{\mathbf{E}_{\bar{C}}(C), C \in \mathcal{C}_p(P_n)\}$$

Proof. Using Lemmas 28 and 34 we can state that $\forall C \in \mathcal{C}_p(P_n) : \mathbf{E}_{\bar{C}}(C) \geq \widehat{\mathbf{E}}_{\bar{C}}(C) \geq \widehat{\mathbf{E}}_{\bar{C}}(B) = \mathbf{E}_{\bar{C}}(B)$

□

The proof of Theorem 30 is hence complete. In the next section, we discuss the case of edge-weighted paths.

4.1.4 The case of edge-weighted path

In this subsection, we show that our approach can not be generalized to edge-weighted paths.

Example. Consider the edge-weighted path G illustrated in Figure 4.4 and 4.5. $B = \{1, 7, 9\}$ where $\lambda(B) = \{\lambda_1^B, \lambda_2^B\} = \{8, 6\}$ is a balanced solution for PpCP with $p = 3$ on G. (as$|\lambda_1^B - \lambda_2^B|$ is minimum, see Definition 23). The value of $r^v(B)$ is given under each vertex v. Note that $r^8(B) = r^9(B) = 4$ due to the evacuation distance of 5. For the other vertices, $r^v(B)$ is equal to the evacuation distance of v or one of its neighbors. Thus, $\mathbf{E}(B) = \sum_{v=1}^{9} \frac{1}{9} r^v(B) = \frac{47}{9}$

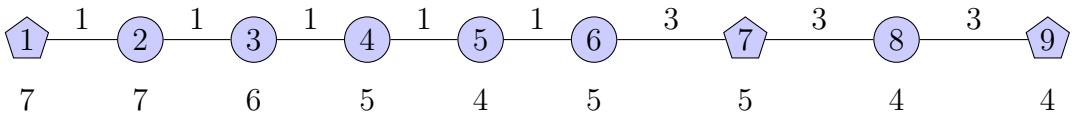


Figure 4.4: The unique balanced solution for PpCP with $p = 3$ on G

Now consider solution $A = \{1, 6, 9\}$ as illustrated in Figure 4.5, with $\lambda(A) = \{\lambda_1^A, \lambda_2^A\} = \{5, 9\}$. As $|\lambda_1^A - \lambda_2^A| = 4$ is not minimum over all possible solutions, A is not a balanced solution. As in Figure 4.4, the value of $r^v(A)$ is given under each vertex v. The value of solution A for the PpCP with $p = 3$ is given by $\mathbf{E}(A) = \sum_{v=1}^{9} \frac{1}{9} r^v(A) = \frac{42}{9} < \mathbf{E}(B)$. Thus B is not the optimal solution.

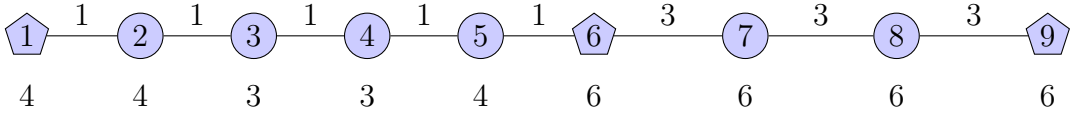


Figure 4.5: A non-balanced solution for PpCP with $p = 3$ on G

♣

We have shown in Theorem 30 that PpCP admits a monotone balanced p-set as an exact solution on paths in the uniform case. However, the approach considered is not valid when we consider edge-weighted paths. While the question of solving

PpCP on acyclic edge-weighted graphs remains open, we propose in the next section an approximation algorithm for PpCP on edge-weighted trees.

4.2 Approximation algorithms

We recall that, for any p-set C of G, the radius of C is $r(C) = \max_{v \in V} d(v, C)$ and $r(C)$ is the value of C for `Min p-Center` on G. Note that for any scenario $s \in V$, $r^s(C) \geq r(C)$.

In this section we will show that, in graphs of bounded average degree, there is a polynomial approximation algorithm guaranteeing a constant approximation ratio for the uniform `PpCP`. Our result is even valid if edge lengths lie into $[l, 2l]$ for a positive l. Our strategy is to show that, under these assumptions, the ratio $\frac{\mathbf{E}(C)}{r(C)}$ is bounded for any p-set C that is feasible for `PpCP`. In particular, a solution with constant approximation ratio for `Min p-Center` has a constant ratio for `PpCP`.

In graphs with general lengths we cannot expect the same as we show it in the next example. Thus, another strategy should be taken.

Example. Consider the caterpillar H of Figure 4.6 with three internal vertices x, y, z and edges (x, y) and (y, z) of length Z and three pending vertices a, b, c, respectively linked to x, y, z with edges of length 1.

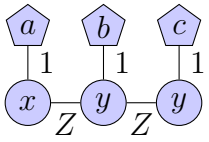


Figure 4.6: A case where $\frac{\mathbf{E}(C)}{r(C)} = Z + 1$.

$\{a, b, c\}$ is the unique feasible solution of the `PpCP` instance $(H, 3)$. We have $r(\{a, b, c\}) = 1$. However, for any scenario s, $r^s(\{a, b, c\}) = Z + 1$, which implies $\mathbf{E}(\{a, b, c\}) = Z + 1$.

♣

Recall that, from Proposition 3 in Section 2.1.5, a set $C \subset V$ is in $\mathcal{C}_p(G)$ if and only if $|C| \geq 2$ and C intersects all minimal articulation components, denoted MACs.

For any $p \leq |V|$, we call *MAC p-set* a p-set intersecting all MACs. For $p \geq 2$, $\mathcal{C}_p(G)$ is then the set of feasible solutions of MAC `Min MAC p-Center`. We consider the `Min MAC p-Center` problem of finding a MAC p-set of minimum radius. The `Min MAC p-Center` problem has a feasible solution for a graph G if and only if p is at least the number of MACs in G, i.e., $p \geq |\Upsilon(G)|$.

`Min MAC p-Center`
Instance: A graph $G = (V, E)$, an integer p
Feasible solutions: $\mathcal{C}_p(G)$ whose elements are called MAC p-set
Objective: Find a MAC p-set of minimum radius $r(C)$.

In the following, to avoid any ambiguity, we will call MAC p-set a p-set taken as a solution for `Min MAC p-Center`.

We outline that a balanced p-set on a path, defined in the previous section, is also an optimal solution for `Min MAC p-Center`. It can be proved through the polynomial algorithm of Kariv and Hakimi [15].

In what follows, we describe an approximation preserving reduction between `PpCP` and `Min MAC p-Center` (Subsection 4.2.1). A polynomial approximation algorithm for the latter leads to a polynomial approximation algorithm for the former with a ratio that depends on the average degree $\overline{deg}(G) = \frac{2|E|}{|V|}$ of G. More precisely, the reduction is even the identity and we analyze how good for the problem `PpCP` an approximated MAC p-set can be. Then, in Subsection 4.2.2, we show that `Min MAC p-Center` can be approximated within the ratio 2, which leads to a $(4\overline{deg}(G) + 2)$-approximation for the uniform `PpCP`. Actually, the result still holds if all edge-lengths lie in the interval $[\ell, 2\ell]$ for any positive ℓ.

4.2.1 A polynomial approximation preserving reduction

We directly establish the following proposition for general edge lengths. We will denote respectively ℓ_M and ℓ_m the maximum and minimum edge lengths.

Proposition 36. On an edge weighted graph with lengths in $[\ell_m, \ell_M]$, $\forall C \in \mathcal{C}_p(G)$, we have:

$$\mathbf{E}(C) \leq (2\overline{deg}(G) + 1)r(C) + (\ell_M - 2\ell_m)\overline{deg}(G)$$

Proof. Let us consider any scenario $s \in V$ of degree $deg(s)$ and number $1, 2, \ldots,$ $deg(s)$ the edges incident to s. We claim that $r^s(C) \leq (2deg(s) + 1)r(C)$.

Consider indeed $x \in V$ such that $r^s(C, x) = r^s(C) \geq r(C)$. If $r^s(C, x) = r(C)$, then the claim is satisfied. Let us assume $r^s(C, x) > r(C)$. We consider two cases:

1. $x \neq s$. $r^s(C, x)$ is the length of a path $\pi = [x_0, x_1, \ldots, x_k]$, where $x_0 = x$, $x_k \in C$ and π is a minimum path in G^s. Since $d^s(x, x_k) > r(C)$, we can define $i = \max\{j \in \{0, \ldots k-1\}, d^s(x_j, x_k) > r(C)\}$. Then all vertices $x_j, j \in \{0, \ldots, i\}$ are, in G, at distance at most $r(C)$ from s. Indeed, the path $x_j, \ldots, x_k$ is a minimum path of length greater than $r(C)$ in G^s. So, in G, the evacuation path of vertices $x_j, j \in \{0, \ldots, i\}$ passes through s. Figure 4.7 illustrates the distance relation between x, s and x_k in the case $x \neq s$. In the figure, no shelter is located on s, but the reasoning is the same if there is one.

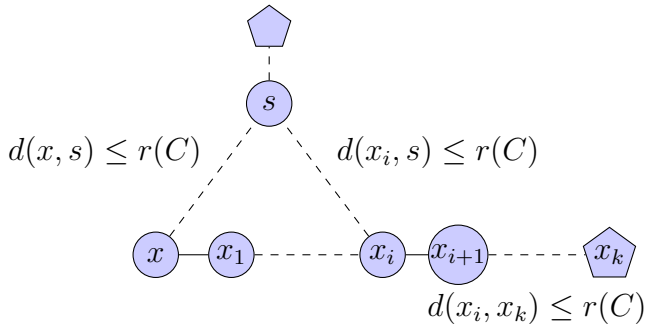


Figure 4.7: Distance relations between vertices x, s and x_k used for Proposition 36

In G, for $j \in \{0,\ldots,i\}$, we consider a minimum path from x_j to s, of value at most $r(C)$. We assign to x_j a color in $N_1,\ldots,N_{deg(s)}$ depending on the last edge of the minimum path we have fixed for x_j: x_j is of color N_t if the related minimum path between x_j and s terminates with the t^{th} edge incident to s.

Note that the distance in G_s between two vertices of the same color is at most $2r(C)-2\ell_m$. Indeed considering, in G, two minimum paths from these vertices to s and sharing the last edge, we deduce a walk avoiding s between them of total length at most $2r(C)-2\ell_m$. This walk includes a path in G^s of length at most $2r(C)-2\ell_m$ between these two vertices.

This allows us to derive an upper bound of $d^s(x,x_i)$. Suppose x is of color N_{i_1} and consider the last vertex x_j of color N_{i_1} along the path π; we have $d^s(x,x_j) \leq 2r(C)-2\ell_m$. Then, if $j<i$, the vertex x_{j+1} is of color N_{i_2} and $d^s(x_j,x_{j+1}) \leq \ell_M$. Using the same reasoning for all non-empty colors gives $d^s(x,x_i) \leq deg(s)(2r(C)-2\ell_m)+(deg(s)-1)\ell_M$.

Taking into account the edge x_ix_{i+1} and the fact that $d^s(x_{i+1},x_k) \leq r(C)$ we have:

$$r^s(C) \leq (2deg(s)+1)r(C)+deg(s)(\ell_M-2\ell_m) \tag{4.8}$$

2. $x=s$ Similarly, $r^s(C,s)$ is the length of a path $\pi=[x_0,x_1,\ldots,x_k]$, where $x_0=s$, $x_k \in C$ and $[x_1,\ldots,x_k]$ is a minimum path in G^s. We define i as in the previous case and use the same argument: x_1 is color N_{i_1} and we define x_j as previously. The only difference is that for any vertex the fixed minimum path from x_j to s passes through x_1 and consequently $d^s(x_1,x_j) \leq r(C)-\ell_m$. For the other colors, the same bound as previously holds. We then get a better

bound:

$$\begin{aligned} r^s(C) \leq & r(C) - \ell_m + (deg(s) - 1)(2r(C) - 2\ell_m) + deg(s)\ell_M + r(C) \\ \leq & 2deg(s)r(C) + deg(s)(\ell_M - 2\ell_m) + \ell_m \end{aligned} \tag{4.9}$$

This bound is better than in Equation 4.8 since $\ell_m \leq r(C)$.

So, in all cases we have $r^s(C) \leq (2deg(s) + 1)r(C)$. We deduce, by taking the average value, $\mathbf{E}(C) = \frac{1}{|V|}\sum_{s\in V} r^s(C) \leq (2\overline{deg}(G) + 1)r(C) + (\ell_M - 2\ell_m)\overline{deg}(G)$ which concludes the proof. □

On a tree, the analysis can be improved:

Proposition 37. On a tree with edge lengths in $[\ell_m, \ell_M]$, $\forall C \in \mathcal{C}_p(G)$, we have:

$$\mathbf{E}(C) \leq 3r(C) + \ell_M - 2\ell_m$$

Proof. Consider, for a scenario s, and a vertex $x, r^s(C, x) = r^s(C)$, the same analysis as in the proof of Proposition 36. Since there is no cycle, all vertices $x, \ldots, x_i$ are of the same color. Equation 4.8 becomes

$$r^s(C) \leq 3r(C) + \ell_M - 2\ell_m$$

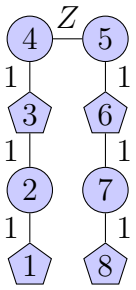
which concludes the proof. □

As noticed in the following example in Figure 4.8, with general weights system the situation may be totally different. In this example, the graph is a path on 8 vertices with only one edge of weight $Z > 1$ and all other edges of weight 1 and $p = 4$. There is a unique optimal MAC 4-center and, for large values of Z, its value is very bad compared to an optimal `PpCP` solution.

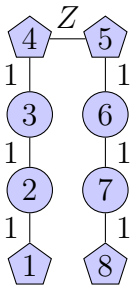
Proposition 38. Suppose a class of edge-weighted graphs $G = (V, E)$ with $\ell_M \leq 2\ell_m$ for which `Min MAC p-Center` can be approximated with $\rho(G)$.
Then, `PpCP` can be approximated with $(2\overline{deg}(G) + 1)\rho(G)$ on the same class.

Proof. Given a graph G in the class, we build a p-set C in $\mathcal{C}_p(G)$, if it exists, of value at most $\rho(G)r^*(G)$, where $r^*(G)$ denotes the optimal radius of a MAC p-set in G. Using Proposition 36 and $\ell_M \leq 2\ell_m$, we have $\mathbf{E}(C) \leq (2\overline{deg}(G) + 1)r(C) \leq (2\overline{deg}(G) + 1)\rho(G)r^*(G)$.

Now if C^* is an optimum solution for `PpCP`, we have $\mathbf{E}(C^*) \geq r(C^*) \geq r^*(G)$. This concludes the proof. □



$\{1,3,6,8\}$ is an optimal MAC 4-center
$r(\{1,3,6,8\}) = 1, \mathbf{E}(\{1,3,6,8\}) = \frac{Z}{2} + 1$



$\{1,4,5,8\}$ is PpCP-optimal for $p = 4$;
$r(\{1,4,5,8\}) = 2, \mathbf{E}(\{1,4,5,8\}) = 2$

Figure 4.8: With general weights, an optimal MAC p-set can be a very bad PpCP solution.

4.2.2 Constant approximation algorithms

Proposition 39. `Min MAC p-Center` is polynomial on edge-weighted trees.

Proof. Given a tree $\mathcal{T}$, for any distance d we consider the tree $\mathcal{T}_d$ obtained from $\mathcal{T}$ by gluing to each pending vertex v a path of length d. Then, $\mathcal{T}$ has a MAC p-set of radius d if and only if $\mathcal{T}_d$ has a p-set of radius d. The result immediately follows from the fact that `Min p-Center` is polynomial on trees. □

Using Proposition 37 and the analysis of Proposition 38, we get:

Corollary 40. There is a polynomial algorithm for PpCP guaranteeing the ratio 3 on trees in the uniform case.

Remark 5. Note however that we leave open the problem of whether PpCP is NP-hard or polynomial on trees.

Next we address the approximation of PpCP on general graphs. In this subsection we will need another variant of `Min p-Center` called `Min Partial p-Center` problem and introduced in [83]. Given an edge-weighted graph $G = (V, E)$ and a set of vertices $U \subset V$, `Min Partial p-Center` is to minimize the *partial radius* $r(C, U)$ of a p-set C, where $r(C, U) = \max_{x \in U} d(x, C)$.

`Min Partial p-Center`

Instance:	An edge-weighted graph $G = (V, E)$, a set $U \subset V$, an integer p
Feasible solutions:	Any p-set
Objective:	Find a p-set C of minimum partial radius.

The underlying logic is that only vertices in U need to be close to a center. However, centers can be any vertex in G and distances are computed in G (within our terminology, it means that evacuation paths are not required to stay in U). Note that, if $U = V$, then $r(C, V) = r(C)$ and `Min Partial p-Center` is just the usual `Min p-Center` problem. So, `Min p-Center` is a particular case of `Min Partial p-Center`.

In particular, `Min Partial p-Center` is not approximable within $2-\varepsilon$ for any $\varepsilon > 0$, unless P=NP by using the same hardness result for `Min p-Center` proved in [84]. Note that this hardness result for `Min p-Center`, directly obtained from the NP-hardness of `Min Dominating Set`, holds in the uniform case. Since `Min Dominating Set` remains NP-hard in planar bipartite graphs of degree 3, `Min p-Center`, and by consequence `Min Partial p-Center`, are not approximable within $2 - \varepsilon$ for any $\varepsilon > 0$ in planar bipartite graphs of degree 3 with all edge lengths 1, unless P=NP. Note that the argument used for `Min Partial p-Center` cannot be easily adapted to `Min MAC p-Center` since this latter problem is not an immediate generalization of `Min p-Center`.

For an edge-weighted graph $G = (V, E)$, we define K the complete edge-weighted graph over V. Let $\tilde{\ell}_{ij}$ be the length of (i, j) in K for all $i, j \in V$, we assume $\tilde{\ell}_{ij} = d(i, j)$

Note that, for `Min p-Center`, the instance (G, p) is equivalent to the instance (K, p). Both instances G and K have the same feasible solutions with the same values and thus, the same optimal solutions. To guarantee finite edge lengths in K, we just consider G is connected. Since K is 2-connected as soon as $|V| \geq 2$, `Min MAC p-Center` is equivalent to `Min p-Center` on K. Since the hardness result for `Min p-Center` still holds in connected graphs, `Min MAC p-Center` is not approximable within $2 - \varepsilon$ for any $\varepsilon > 0$, unless P=NP. We can even easily show that this hardness results already holds for the uniform case. To this aim, we just need to show that `Min Dominating Set` is NP-hard in 2-connected graphs. Given a graph $G = (V, E)$ instance of `Min Dominating Set`, we construct G' from G as follows: for every articulation point a of G, create a twin vertex a' linked to a and to all neighbors of a. G' is 2-connected and the `Min Dominating Set` problems in G and G' are equivalent. Now, a set of p vertices in G' is a dominating set if and only if its radius is 1 and else, the minimum radius of a p-set is at least 2. It immediately implies that `Min MAC p-Center` in graphs with edge lengths all equal to 1 is not approximable within $2 - \varepsilon$ for any $\varepsilon > 0$, unless P=NP.

In what follows, we propose a polynomial 2-approximation algorithms for `Min Partial p-Center` and `Min MAC p-Center`. These approximation results hold even in the case with general lengths.

Consider an instance (G, U) of `Min Partial p-Center`, where $G = (V, E)$ is an edge-weighted graph and $U \subset V$. We recall $n = |V|$. We can compute K in $O(n^3)$. We denote $SL = \{d(x, y), x, y \in V\}$ the set of edge lengths in K (note that $|SL| \leq n^2$) and for any $d \in SL$, $K_d = (V, E_d)$ is the partial graph of K where E_d is the set of edges of length at most d. Note that for any solution of `Min p-Center`, its radius is in SL.

Theorem 41 can be obtained using the general method in [85] or by adapting the 2-approximation algorithm for `Min p-Center` in [44]. Since it cannot be deduced from existing results, we give a direct proof. We first introduce the main concepts and claims used for this result since they are used as well in Theorem 42.

A solution to `Min p-Center` of partial radius d in (G, U) can be seen as a *partial dominating set* of (K_d, U), where a partial dominating set X is a set of vertices such that every vertex in U has at least one neighbor in X. If A_d is the adjacency matrix of K_d with additional 1s on the diagonal (alternatively A_d is the adjacency matrix of K_d with additional loops on each vertex), we denote $A_{d,U}$ the sub-matrix of A_d corresponding to rows in U (it has $|U|$ rows and $|V|$ columns). The problem of finding a minimum partial dominating set can then be formulated by the following mathematical program $PDS(G, U, d)$, where the 1s on the diagonal represent the fact that a vertex dominates itself:

$$PDS(G,U,d): \begin{cases} \min & \langle \mathbf{1}_{|V|}, x \rangle \\ & A_{d,U}x \geq \mathbf{1}_{|U|} \\ & x \in \{0,1\}^{|V|} \end{cases}$$

We then consider the mathematical program $SIS(G, U, d)$ that corresponds to finding a maximum *strong independent set* of K_d contained in U, where a strong independent set $S \subset V$ is an independent set (every two vertices in S are not adjacent) such that every vertex in $V \setminus S$ has at most one neighbor in S.

$$SIS(G,U,d): \begin{cases} \max & \langle \mathbf{1}_{|U|}, y \rangle \\ & A_{d,U}^{\intercal} y \leq \mathbf{1}_{|V|} \\ & y \in \{0,1\}^{|U|} \end{cases}$$

Claim 3. The cardinality of any strong independent set of K_d contained in U is not more than the cardinality of any partial dominating set of (K_d, U).

Proof. The relaxations of mathematical programs $PDS(G, U, d)$ and $SIS(G, U, d)$, replacing the binary conditions with non negative conditions, are dual linear programming problems. The result is an immediate consequence of the weak duality theorem. □

Let $d_{max} = \max(SL)$. We denote $K_{2d,U}$ the graph $K_{\min(2d,d_{max})}[U]$.

Claim 4. For a given distance $d \in SL$, let S_d be a maximal independent set of $K_{2d,U}$. S_d is a partial $|S_d|$-center in (G, U) of partial radius $r(S_d, U) \leq 2d$.

Proof. Consider any vertex $u \in U \setminus S_d$. Since S_d is maximal, $S_d \cup \{u\}$ is not independent in $K_{2d,U}$, which means $d(u, S_d) \leq 2d$ and the claim is proved. □

Claim 5. Any independent set S of $K_{2d,U}$ is a strong independent set of K_d contained in U.

Proof. By definition, $S \subset U$. Since S is independent in $K_{2d,U}$, it is independent in $K_{d,U}$, a partial graph of $K_{2d,U}$. So, it is an independent set of K_d. The result then follows by contrapositive: if there is a vertex $u \in V \setminus S$ adjacent, in K_d, to two vertices of S, then these two vertices would be at distance at most $2d$, so would be adjacent in $K_{2d,U}$. □

Claims 3, 4 and 5 immediately allow to derive an approximation algorithm for `Min Partial p-Center`.

Theorem 41. `Min Partial p-Center` is polynomially 2-approximable and this is the best possible constant ratio.

Proof. We already noted that 2 is a lower bound for any constant approximation ratio of `Min Partial p-Center`. So, we only need to prove that this bound can be guaranteed.

For a given instance (G, U), we can compute SL and all distances $d(i, j), i, j \in V$ in $O(n^3)$. Then, for any $d \in SL$, we can compute a maximal independent set S_d of $K_{2d,U}$ and then select $S_{\tilde{d}}$, where $\tilde{d} \in \underset{d \in SL, |S_d| \leq p}{\operatorname{argmin}} (r(S_d))$. In other words, $S_{\tilde{d}}$ is of minimum value among all S_ds of cardinality at most p. Denote r^*_U the minimum partial radius of a p-set in (G, U). $r^*_U \in SL$. Using Claim 5 and Claim 3, $|S_{r^*_U}| \leq p$ and thus, $\tilde{d}$ exists and $r(S_{\tilde{d}}) \leq r(S_{r^*_U})$. Using Claim 4, we deduce $r(S_{r^*_U}) \leq 2r^*_U$, which completes the proof. □

Note that, using a binary search on the same model as the 2-approximation algorithm for `Min p-Center` proposed in [44], we can design a 2-approximation algorithm of complexity $O(n^2 \log n)$ as soon as all distances between two vertices in G are computed.

We use similar ideas and the same claims to derive a polynomial 2-approximation algorithm for `Min MAC p-Center` (Algorithm 1).

To simplify the description of Algorithm 1, we introduce some notations used in the description of the algorithm. Given the instance $G = (V, E)$, we denote by k the number of MACs of G. These MACs are denoted $A_1, \ldots A_k$ and the related articulation points are called $a_1, \ldots a_k$ (we may have $a_i = a_j, i \neq j$). As previously $SL = \{d(i, j), i, j \in V, \}$; for any $d \in SL$, we partition $I = \{1, \ldots, k\}$ into $I =$

Algorithm 1 2-approximation for `Min MAC p-Center`.

Require: Edge-weighted graph $G = (V, E)$ and $p \geq 2$.
Ensure: Outputs C, a MAC p-set if it exists.

Begin
Compute $A_1, \dots A_k$, and $a_1, \dots a_k$
if $k > p$ **then**
  No-solution output
else
  Compute SL and all distances $d(i,j), i,j \in V$
  $\widetilde{SL} \leftarrow \emptyset$
  for $d \in SL$ **do**
    Compute I_d^- and I_d^+
    $C_d \leftarrow \emptyset$
    for $i \in I_d^-$ **do**
      Select $x \in A_i$
      $C_d \leftarrow C_d \cup \{x\}$
    end for
    $V'_d \leftarrow \{v \in V, d(v, \{a_i, i \in I_d^-\}) > d\}$
    $S_d \leftarrow \emptyset$
    for $i \in I_d^+$ **do**
      Select $y \in \underset{x \in A_i}{\text{argmax}}\ d(x, a_i)$
      $S_d \leftarrow S_d \cup \{y\}$
    end for
    while $\exists v \in V'_d, d(v, S_d) > 2d$ **do**
      $S_d \leftarrow S_d \cup \{v\}$
    end while
    if $|S_d| \leq p - |I_d^-|$ **then**
      $\widetilde{SL} \leftarrow \widetilde{SL} \cup \{d\}$
      $C_d \leftarrow C_d \cup S_d$
    end if
  end for
  Let $\tilde{d} \in \underset{d \in \widetilde{SL}}{\text{argmin}}\ (r(C_d))$
  $C \leftarrow C_{\tilde{d}}$
  return C
end if
End

$I_d^- \sqcup I_d^+$ ($\sqcup$ denotes the disjoint union), where $I_d^- = \{i \in I, \max_{x \in A_i} d(x, a_i) \leq d\}$ and $I_d^+ = \{i \in I, \max_{x \in A_i} d(x, a_i) > d\}$. MACs $A_i, i \in I_d^-$ are seen as small MACs relative to d, while MACs $A_i, i \in I_d^+$ are seen as large ones. "No-solution output" is any output we use to indicate that the problem has no feasible solution.

The idea of the Algorithm is as follows:

1. If the number of MAC is more than p, then there is obviously no solution.
2. Else, for every distance $d \in SL$, Algorithm 1 tries to compute a MAC p-set C_d of radius at most $2d$; only feasible MAC p-set obtained through this process will be kept and $\widetilde{SL}$ is the set of distances d for which it will occur;
3. C_d is built as follows:
   (a) The algorithm selects one center per small MAC $A_i, i \in I_d^-$;
   (b) For each $i \in I_d^-$, all vertices at distance at most d from a_i are allocated to the related center (by definition of I_d^-, this includes in particular all vertices of A_i).
   (c) V'_d is the set of uncovered vertices. If possible, the algorithm completes C_d with a partial $(p - |I_d^-|)$-center of $(G \setminus \bigcup_{i \in I_d^-} A_i, V'_d)$ of partial radius at most $2d$. To this aim, it uses the same ideas as in Theorem 41: it constructs a maximal independent set S_d of K_{2d,V'_d}, but to ensure it intersects all A_is, $i \in I_d^+$, it initializes it by choosing one vertex in each of these components. If $|S_d| \leq p - |I_d^-|$, then $d \in \widetilde{SL}$;
4. The best solution $C_{\tilde{d}}, d \in \widetilde{SL}$ is selected as an approximated solution for `Min MAC p-Center`.

Theorem 42. Algorithm 1 is a polynomial 2-approximation algorithm for `Min MAC p-Center` and this is the best possible constant ratio.

Proof. We already noted that 2 is a lower bound for constant approximation ratios. So, we only need to prove that this bound can be guaranteed.

Assume that $k \leq p$; then the instance of `Min MAC p-Center` has feasible solutions and thus, also an optimal solution.

Fix a distance $d \in SL$. Note first that, by definition of I_d^- and I_d^+, V'_d computed at line 15 satisfies $V'_d \subset V \setminus \bigcup_{i \in I_d^-} A_i$ and $\forall i \in I_d^+$, $A_i \cap V'_d \neq \emptyset$. Then, the algorithm computes the set S_d from Lines 16 to Line 23.

Claim 6. $\forall d \in SL$, S_d is a maximal independent set in K_{2d,V'_d} that intersects all A_is, $i \in I_d^+$.

Proof. The algorithm initializes S_d by selecting, in each MAC $A_i, i \in I_d^+$, a vertex at maximum distance from a_i. This ensures that, at Line 20, S_d includes one element per MAC $A_i, i \in I_d^+$ and is an independent set (possibly empty) in K_{2d,V'_d}. Indeed, if y_i, y_j are respectively selected at Line 18 for $i, j \in I_d^+, i \neq j$, then any path between them passes through a_i and a_j (we may have $a_i = a_j$) and is of length greater than $2d$. As a consequence, S_d is a maximal independent set in K_{2d,V'_d}. □

$\widetilde{SL}$, computed by the algorithm (Lines 25), is the set of distances d such that S_d is of size at most $p - |I_d^-|$. Consider now an optimal MAC p-set, C^*_{MAC}, of radius d^*.

Claim 7. $d^* \in \widetilde{SL}$

Proof. Since C^*_{MAC} has at least one center per MAC, C^*_{MAC} has at most $\left(p - |I_{d^*}^-|\right)$ centers in $V \setminus \bigcup_{i \in I_{d^*}^-} A_i$. In addition, vertices in V'_{d^*} cannot be associated with (i.e., evacuated to) centers in $\bigcup_{i \in I_{d^*}^-} A_i$ since these centers are at distance more than d^*. This means that $C^*_{MAC} \cap (V \setminus \bigcup_{i \in I_{d^*}^-} A_i)$ is a $(p - |I_{d^*}^-|)$-center of partial radius at most d^* in $(G \setminus \bigcup_{i \in I_{d^*}^-} A_i, V'_{d^*})$.

As a consequence $C^*_{MAC} \cap (V \setminus \bigcup_{i \in I_{d^*}^-} A_i)$ is a partial dominating set in (K_{d^*}, V'_{d^*}). Using Claims 5 and 3, we get $|S_{d^*}| \leq |C^*_{MAC} \cap (V \setminus \bigcup_{i \in I_{d^*}^-} A_i)| \leq p - |I_{d^*}^-|$, which means $d^* \in \widetilde{SL}$. □

Claim 7 ensures in particular that $\widetilde{SL} \neq \emptyset$ and consequently $\tilde{d}$ computed at Line 29 is well defined. Since d^* and $\tilde{d}$ are both in $\widetilde{SL}$, the algorithm computes both sets C_{d^*} and $C_{\tilde{d}}$ by selecting one vertex per $A_i, i \in I_{d^*}^-$ and one vertex per $A_i, i \in I_{\tilde{d}}^-$, respectively (from Line 10 to Line 14) and completing with S_{d^*} and $S_{\tilde{d}}$, respectively. Using Claim 6, this ensures that both C_{d^*} and $C_{\tilde{d}}$ are MAC p-set.

Finally, $C_{\tilde{d}}$ is selected as approximated solution and Line 29 ensures

$$r(C_{\tilde{d}}) \leq r(C_{d^*}) \tag{4.10}$$

We complete the proof by showing the following claim.

Claim 8. $r(C_{d^*}) \leq 2d^*$.

Proof. Consider first a vertex $v \in V'_{d^*}$ and use the same argument as in the proof of Theorem 41. We have $d(v, C_{d^*}) \leq d(v, C_{d^*} \setminus \bigcup_{i \in I_{d^*}^-} A_i) \leq r(S_{d^*}, V'_{d^*})$. Using Claims 6 and 4, we have $r(S_{d^*}, V'_{d^*}) \leq 2d^*$ and thus:

$$\forall v \in V'_{d^*}, d(v, C_{d^*}) \leq 2d^*. \tag{4.11}$$

Consider now a vertex $v \in V \setminus V'_{d^*}$. By definition of V'_{d^*}, we have $d(v, \{a_i, i \in I^-_{d^*}\}) \leq d^*$ and by definition of $I^-_{d^*}$, we have $\exists i \in I^-_{d^*}, \forall u \in A_i, d(v,u) \leq 2d^*$. This ensures:

$$\forall v \in V'_{d^*}, d(v, C_{d^*}) \leq 2d^*. \tag{4.12}$$

Equations 4.11 and 4.12 ensure $r(C_{d^*}) \leq 2d^*$. □

Claim 8 and Equation 4.10 imply $r(C_{\tilde{d}}) \leq 2d^*$, which concludes the proof of Theorem 42.

□

We immediately deduce from Theorem 41 and Proposition 38:

Corollary 43. For edge weighted graphs with lengths in $[\ell, 2\ell]$, PpCP is approximable within $4\overline{deg}(G) + 2$.

Conclusion

In this chapter, we show in Section 4.1 that, on a path, and in the uniform case, a balanced monotone solution is an optimal solution for PpCP (Theorem 30). It requires to decompose the problem into subproblems. We introduce, for each scenario, the local evacuation radius as the maximum evacuation distance of the source vertex and its neighborhood. The local evacuation radius is a lower bound to the evacuation radius of a solution. This bound is tight when the solution is balanced. In addition, we consider a partition of the scenarios in two sets. For each set of scenarios, we prove independently that a balanced monotone solution minimizes the local evacuation radius.

In Section 4.2, we propose some approximation results for PpCP. In edge weighted graphs of bounded degree and with edge lengths in $[\ell, 2\ell]$, PpCP is approximable within the constant ratio $4\overline{deg}(G) + 2$. On trees, PpCP is approximable within the constant ratio 3. This approximation result holds for a class of instances on which all our hardness results apply: it provides a first gap between constant approximation ratios and the hardness in approximation results we have obtained.

In addition, we introduce the Min MAC p-Center problem, which is the problem of finding a p-set of minimum radius among the set of feasible solutions for PpCP. We show that this problem is 2-approximable and that this is the best possible constant approximation ratio. It is also polynomial on trees.

In the next chapter, we consider the RpCP problem where we generalize some notions introduced in this chapter. Indeed, the idea of a segment-partition induced

by a solution will be used and generalized to develop polynomial algorithms for `RpCP` on edge-weighted acyclic graphs.

CHAPTER 5

POLYNOMIAL AND APPROXIMATION RESULTS FOR RpCP

Abstract

In this chapter we present polynomial exact algorithms for `RpCP` on edge-weighted acyclic graphs. A generic algorithm is presented, based on a binary search coupled with the auxiliary problem `k-RCP`. The complexity of the algorithm mainly depends on the complexity of an algorithm for `k-RCP`. For this purpose, we propose efficient algorithms to solve `k-RCP` in polynomial time on paths, extended stars, caterpillars and trees.

Contents

5.1 The general approach 98

5.2 Edge-weighted paths 100

5.2.1 Computing the robust radius of a p-set 100

5.2.2 Bounds for `RpCP` 101

5.2.3 Algorithm 101

5.2.4 Example 103

5.3 Extended star with edges of different lengths 105

5.3.1 Computing the robust radius of a p-set 106

5.3.2 Algorithm 107
5.4 Caterpillar 108
5.4.1 Computing the robust radius of a p-set 109
5.4.2 A basic algorithm for k-RCP 110
5.4.3 An improved algorithm for k-RCP 112
5.5 Trees 116
5.5.1 Computing the robust radius of a p-set 116
5.5.2 Algorithm for k-RCP 117

Introduction

We have seen in the Chapter 3 that RpCP is NP-hard on bipartite planar graphs and subgrids of degrees $\{2, 3\}$. In this chapter, we propose polynomial algorithms to solve RpCP on several simple classes of graphs.

In Section 5.1, we present a generic algorithm to solve RpCP using a binary search with k-RCP. We show that RpCP can be solved in polynomial time on a graph if k-RCP can be solved in polynomial time on the same graph. In Section 5.2, we propose a polynomial algorithm in $O(n)$ to solve k-RCP on edge-weighted paths. In Section 5.3, we propose a polynomial algorithm in $O(n)$ to solve k-RCP on edge-weighted extended stars. In Section 5.4, we propose a polynomial algorithm to solve k-RCP on an edge-weighted caterpillar. We present also an improved algorithm that narrows the complexity of the problem to $O(n)$. In Section 5.5, we propose a polynomial algorithm to solve efficiently k-RCP on edge-weighted trees. Our algorithm is based on observations and structural properties inferred from the previous cases. We propose an efficient algorithm to solve k-RCP in $O(n)$.

5.1 The general approach

In Section 2.1.4, we introduced k-RCP as a problem closely related to RpCP. On an edge-weighted graph and for a given k, k-RCP is the problem of determining a set of vertices of minimum size such that its robust radius is less than or equal to k. We present in this section a generic algorithm for RpCP based on k-RCP.

Consider an instance (G, p) of RpCP, with $p \geq 2$ and G an edge-weighted graph. We recall that D is the ordered set of candidate values for the robust radius of G i.e., $D = \{d^s(x, y) : x, y, s \in V\} \cup \{l_{sx} + d^s(x, y) : y, s \in V, x \in \Gamma(s)\}$.

We observe that, for $k \in D$, if the optimal value of k-RCP on (G, k) is greater than p, then the optimal value of RpCP is greater than k. Else, the optimal value of RpCP is less or equal to k. Based on this relation, we can perform a binary search on D to find the minimum radius k^* for which the optimal value of k-RCP is less or equal to p. Note that, if lower and upper bounds are known for the minimum robust radius of G, then we can restrict D before starting the binary search. We implement the method we just described in Algorithm 2.

Algorithm 2 Exact algorithm for RpCP

Require: Graph G, integer p
Ensure: An optimal solution C for RpCP on G
1: Generate the increasing ordered set D.
2: Restrict D with known lower and upper bounds.
3: **while** $|D| > 1$ **do**
4: Let k be the median value of the restricted D.
5: Solve $\mathtt{k-RCP}$ with (G, k). Denote C the obtained solution and p' its size.
6: If $p' \leq p$, restrict D to its first half. Otherwise restrict to its second half.
7: **end while**

The output of Algorithm 2 is a solution C of size at most p and of radius k. The resulting C is an optimal solution for RpCP as no solution of size p exists with a smaller radius.

The complexity of generating D is given by the complexity of generating the shortest path between every pair of nodes in G^s, for every scenario $s \in V$. On general graphs, for every scenario s, there are at most $O(n^2)$ shortest paths which can be computed in $O(nm)$ ([41]). As there are n scenarios, the overall complexity of generating D is $O(n^2m)$, with $|D| \leq n^3$. On specific classes of graphs, the complexity of generating D can be improved. For example, on connected acyclic graphs, there is exactly one path between two vertices. Therefore, the set of evacuation paths is contained in the set of shortest paths and D can be obtained by computing the lengths of all the shortest paths in G. Thus, on all acyclic graphs, as $|E| < n$, D can be computed in $O(n^2)$ and is of maximum size n^2.

In all cases, the while-loop need to be repeated at most $\log(|D|) = O(\log(n))$. Finally, the complexity of solving k-RCP depends on the considered class of the graph. In summary, on a class of graph $\mathbb{H}$, if D can be generated in $O(g(n))$ and k-RCP can be solved in $O(f(n))$, then RpCP can be solved in $O(g(n))+O(f(n)\log n)$ with Algorithm 2. Hence, developing a polynomial algorithm for k-RCP would ensure a polynomial algorithm for RpCP. It follows that k-RCP is NP-hard on the classes of graphs on which RpCP is NP-hard. In the next sections, we consider

that D is given and we propose polynomial algorithms for `k-RCP` on some specific edge-weighted acyclic graphs. We start with paths.

5.2 Edge-weighted paths

Let G be an edge-weighted path. We apply the notations introduced in Section 4.1.1. We recall that, for a given solution $C = \{c_1, \ldots, c_q\} \subset V$, we denote c_i, c_{i+1} the extremities of μ_i^C, where $i \in \{1, \ldots, q-1\}$. We denote a_i and b_i the vertices of μ_i^C adjacent to c_i and c_{i+1} respectively. In addition, we denote $\alpha_i^C = l_{a_i c_i}$, $\beta_i^C = l_{b_i c_{i+1}}$, and ϕ_i^C such that $\alpha_i^C + \phi_i^C + \beta_i^C = \lambda_i^C$. We illustrate these notations on Figure 5.1.

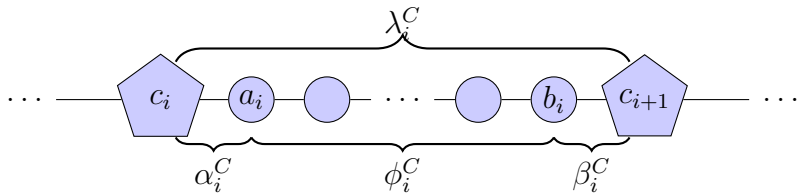


Figure 5.1: Values λ_i^C, α_i^C, ϕ_i^C and β_i^C illustrated on a segment μ_i^C

On a path G, for a given feasible solution C of `k-RCP`, every internal vertex has exactly two potential evacuation paths. Indeed, let c_i, c_{i+1} be the nearest shelters on both sides of a vertex x, such that $c_i < x < c_{i+1}$. The evacuation path of x will always be one of the two paths P_{xc_i} or $P_{xc_{i+1}}$. If a_i and b_i do not exist, then we set $\alpha_i^C = \lambda_i^C$ and $\beta_i^C = 0$.

In the following, $\forall x, y \in V, x < y$, we say x is located downstream of y, and y is located upstream of x. The direction from x to y is called upstream, the opposite direction is called downstream.

5.2.1 Computing the robust radius of a p-set

Proposition 44. On a path, the robust evacuation radius of a feasible solution C for `k-RCP` on G is:

$$\mathrm{R}(C) = \max_{i \in \{1,\ldots,q-1\}} \{\phi_i^C + \beta_i^C, \phi_i^C + \alpha_i^C\} \tag{5.1}$$

Proof. Suppose $j \in \mu_i$ with c_i, c_{i+1} the extremities of μ_i. If $j \in C$, then $r^s(C, j) = 0$ for any scenario s, else we observe that the evacuation path of j in any scenario is contained in one of the evacuation paths of a_i or b_i for scenario a_i or b_i respectively. Thus, $r^s(C, j) = \max\{r^{a_i}(C, a_i), r^{b_i}(C, b_i)\}$. It follows that:

$$\begin{aligned}
\mathbf{R}(C) &= \max_{i\in\{1,\dots,q-1\}}\{r^{a_i}(C,a_i), r^{b_i}(C,b_i)\} \\
&= \max_{i\in\{1,\dots,q-1\}}\{l_{c_ia_i}, \lambda_i^C - l_{c_ia_i}, l_{b_ic_{i+1}}, \lambda_i^C - l_{b_ic_{i+1}}\} \\
&= \max_{i\in\{1,\dots,q-1\}}\{\alpha_i^C, \lambda_i^C - \alpha_i^C, \beta_i^C, \lambda_i^C - \beta_i^C\} \\
&= \max_{i\in\{1,\dots,q-1\}}\{\alpha_i^C, \phi_i^C + \beta_i^C, \beta_i^C, \phi_i^C + \alpha_i^C\} \\
&= \max_{i\in\{1,\dots,q-1\}}\{\phi_i^C + \beta_i^C, \phi_i^C + \alpha_i^C\}
\end{aligned}$$

□

For a given C, α_i^C, ϕ_i^C and β_i^C can be computed for all values of i by scanning all vertices once. Thus $\mathbf{R}(C)$ can be computed in $O(n)$.

5.2.2 Bounds for RpCP

A lower bound for k can be computed from p. For any solution C of size p, $\alpha_i^C + \phi_i^C \leq k$ and $\phi_i^C + \beta_i^C \leq k \; \forall i \in \{1, \dots, p-1\}$. Thus $\lambda_i^C \leq \alpha_i^C + 2\phi_i^C + \beta_i^C \leq 2k$. p is minimal if all the segments are of maximal length $\lambda_{max}^C = 2k$, in which case there is at least $\lceil \frac{\sum_{i=1}^{n-1} l_{i,i+1}}{\lambda_{max}^C} \rceil$ segments i.e., $p - 1 \geq \frac{\sum_{i=1}^{n-1} l_{i,i+1}}{2k} \Leftrightarrow k \geq \frac{\sum_{i=1}^{n-1} l_{i,i+1}}{2(p-1)}$. Thus, for a given p, when solving RpCP on G, D can be restricted to the set of its values greater or equal to $\frac{\sum_{i=1}^{n-1} l_{i,i+1}}{2(p-1)}$.

An upper bound can be obtained from any randomly computed feasible solution. In the case of a path, a balanced solution (introduced in Section 5.2) can be easily computed and induces an upper bound.

5.2.3 Algorithm

Next we present an algorithm to compute an optimal solution for k-RCP on paths. The idea of the algorithm is to locate shelters greedily as upstream as possible. By doing so, we show that we minimize the number of centers.

In Algorithm 3, for a given C of size q, α, ϕ and β correspond respectively to $\alpha_{q-1}^C, \phi_{q-1}^C$ and β_{q-1}^C. In other words, α, ϕ and β characterize the last (as in located the most upstream) segment induced by C.

Proposition 45. Algorithm 3 returns an optimal solution of k-RCP for a path and an integer k.

Algorithm 3 Solving `k-RCP`- Paths

Require: Edge-weighted path G, integer k
Ensure: Returns an optimal `k-RCP` solution C for G
1: Add $\{1, 2\}$ to C
2: $\alpha \leftarrow l_{1,2}$, $\phi \leftarrow 0$, $\beta \leftarrow 0$
3: **for** $v \in [3, \ldots, n]$ **do**
4: Add $\{v\}$ to C and remove $\{v-1\}$ from C
5: $\phi \leftarrow \phi + \beta$
6: $\beta \leftarrow l_{v-1,v}$
7: **if** $(\phi + \alpha > k)$ OR $(\phi + \beta > k)$ **then**
8: Add $\{v-1\}$ to C
9: $\alpha \leftarrow l_{v-1,v}$
10: $\phi \leftarrow 0$
11: $\beta \leftarrow 0$
12: **end if**
13: **end for**
14: Return C

Proof. We denote by $G(v)$ the subgraph $G[\{1, 2, \ldots, v\}]$, and by $C(v)$ the state of solution C at the end of iteration v of the loop (steps 3-13). We define similarly $\alpha(v), \phi(v)$ and $\beta(v)$.

The statement to prove is $P(v)$: " $C(v)$ is an optimal solution for $G(v)$". A secondary statement useful for our proof is $P'(v)$: "The centers of $C(v)$ are located maximally upstream in $G(v)$". More precisely, $P'(v)$ states that, if any shelter is moved upstream in $G(v)$, then $\mathbf{R}(C(v)) > k$. We prove by induction that, $\forall v \in \{2, n\}$, $P(v)$ and $P'(v)$ are true.

Base case The base case is for $v = 2$. C is initialized with $C(2) = \{1, 2\}$ which is the trivial optimal solution for $G(2)$. Thus $P(2)$ and $P'(2)$ are trivially true.

Inductive step Assume $P(v)$ and $P'(v)$ are true at the end of iteration v. We will prove that $P(v+1)$ and $P'(v+1)$ are true at the end iteration $v+1$. By step 4, $C(v+1) = C(v) \setminus \{v\} \cup \{v+1\}$. Note that, all segments, except the one located the most upstream, are unaffected by this change. So are the evacuation paths of these segments. As the last segment induced by $C(v+1)$ has been extended with edge $(v, v+1)$, ϕ and β are updated accordingly. We then verify whether $(\phi(v+1) + \beta(v+1))$ or $(\phi(v+1) + \alpha(v+1))$ exceeds k. There are two cases:

- if $\phi(v+1)+\beta(v+1)$ and $\phi(v+1)+\alpha(v+1)$ do not exceed k, then $C(v+1)$ is a feasible solution. As $|C(v+1)| = |C(v)|$, then, based on $P(v)$, $C(v+1)$ is

of optimal size. Hence, $P(v+1)$ is true. In addition, as the most upstream shelter has been relocated to the upstream extremity of the path, $P'(v+1)$ is true.

- if $\phi(v+1)+\beta(v+1)$ or $\phi(v+1)+\alpha(v+1)$ exceeds k, then by Proposition 44, $\mathbf{R}(C(v+1)) > k$. In this case we add vertex v to $C(v+1)$. As $C(v) \subset C(v+1)$, $C(v+1)$ ensures evacuation distances less or equal to k on $G(v)$. Hence $C(v+1)$ is feasible. As the number of shelters of $C(v)$ were minimum ($P(v)$) and located as upstream as possible ($P'(v)$), no shelter in $G(v)$ can be relocated to an upstream vertex without breaking the feasibility condition. It follows that adding a shelter was necessary to satisfy the feasibility condition, thus $P(v+1)$ is true. By $P(v)$, the centers of $C(v)$ are located maximally upstream in $G(v)$. As $C(v) \subset C(v+1)$ and $(v+1) \in C(v+1)$, then the centers of $C(v+1)$ are located maximally upstream in $G(v+1)$. Thus $P'(v+1)$ is true.

Conclusion By induction, $P(v)$ and $P'(v)$ are true for all $v \in \{2, n\}$.

The algorithm ends after less than n iterations and returns $C(n)$, which is an optimal solution for `k-RCP` on $G(n) = G$.

□

The complexity of Algorithm 3 is induced by the number of iterations, that is $n-3$, thus Algorithm 3 is in $O(n)$. Note that, in Algorithm 3, the size of a solution never decreases between two consecutive iterations. For this reason, while building a solution for `k-RCP` with Algorithm 3 in step 5 of Algorithm 2, if the size of the solution exceeds p at one point, then we can proceed to the next step of Algorithm 2.

5.2.4 Example

We look for the `RpCP` for $p = 3$ solution on the edge-weighted graph $G = (V, E)$ with 9 vertices in Figure 5.2 (similar to Section 4.1.4). We apply Algorithm 2 combined with Algorithm 3 on G.

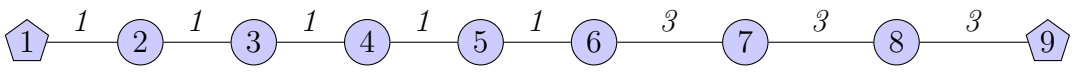


Figure 5.2: Example graph on which we apply Algorithm 2 for $p = 3$

The set D is the range of integer values from 1 to 14. A lower bound for k is given by $\frac{14}{2(p-1)} = \frac{14}{4} = 3.5$. In addition, an upper bound to k can be obtained from

Iteration v	2	3	4	5	6	7	7	8	9	9
$C(v)$	1,2	1,3	1,4	1,5	1,6	1,7	1,6,7	1,6,8	1,6,9	1,6,8,9
α	1	1	1	1	1	1	3	3	3	3
ϕ	0	0	1	2	3	4	0	0	3	0
β	0	1	1	1	1	3	0	3	3	0
$\alpha + \phi$	1	1	2	3	4	5	3	3	6	3
$\phi + \beta$	0	1	2	3	4	7	0	3	6	0
Feasibility	Y	Y	Y	Y	Y	N	Y	Y	N	Y

Table 5.1: Description of the evolution of the parameters of the Algorithm 3 applied to G.

a balanced solution. As we have seen in Section 4.1.4, G has a balanced solution of robust radius 7. It follows that $D = \{4, 5, 6, 7\}$, and its median value is 5. Then, as set out in Algorithm 2, we solve `k-RCP` for $k = 5$ on G using Algorithm 3. Table 5.1 gives a detailed description of the state of the algorithm parameters at step 7 - and step 12 if appropriate - at each iteration. The first column lists the described parameters. The iterations are labeled by the value of v. At iteration 2, we initialize the solution with $\{1, 2\}$, the value of α with $l_{1,2}$, and the values of ϕ and β with 0. As $\alpha + \phi = 1$ and $\beta + \phi = 0$ are less than 5, $C(2)$ is feasible for $G(2)$. At iteration 3, we remove 2 from the solution C and add 3 to it. We update ϕ with $\phi + \beta$ first, then we set the new value of β to $l_{2,3} = 1$. As $\alpha + \phi = 1$ and $\phi + \beta = 1$, $C(3)$ is feasible for $G(3)$. We repeat this procedure for each vertex till iteration 7. For $C(7) = 7$, ϕ is updated with the value $\beta + \phi = 4$, then we set the new value of β to $l_{6,7} = 3$. It follows that $\alpha + \phi = 5$, and that $\beta + \phi = 7$. As the latter sum exceeds 5, $C(7) = \{1, 7\}$ is not a feasible solution for $G(7)$. Thus, we reinsert 6 in $C(7)$, and we fix the value of α to $l_{6,7} = 3$, and the values of ϕ and β to 0. Then $\alpha + \phi = 3$ and $\beta + \phi = 0$ do not exceed 5, thus $C(7) = \{1, 6, 7\}$ is a feasible solution on $G(7)$.

The output solution C is given by the feasible solution $C(9) = \{1, 6, 8, 9\}$ in the last column. We observe that $C(9)$ is of size 4, which is higher than $p = 3$. Before proceeding, we will show the solution if we had started the algorithm with the other extremity of the path, namely with $v = 9$. To this end, consider the graph G', isomorph to G, such that the vertex numbers are inversed. We observe that the optimal solution obtained $C' = \{1, 3, 7, 9\}$ does not correspond to solution C in G, however C' is also of size 4.

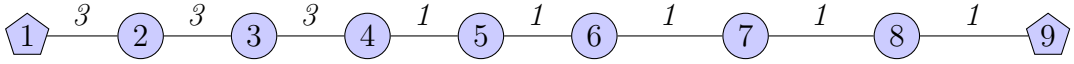


Figure 5.3: Example graph G' on which we apply Algorithm 2 for $p = 3$

Thereby we know that there is no solution for RpCP for $p = 3$ on G of radius less or equal to 5, then 5 is a new strict lower bound on k. Hence we proceed to a new iteration in Algorithm 2 with $D = \{6, 7\}$. We apply Algorithm 3 on G with $k = 6$, which gives us the solution $\{1, 5, 9\}$. It is optimal on G.

In the next section, we consider the class of extended stars.

5.3 Extended star with edges of different lengths

Consider S_t an *extended star* of n vertices, that is a tree with t pending vertices and all the internal vertices of degree two, except one internal vertex of degree t, called the *root* and denoted o. We call *branch* the path between o and a pending vertex, both included. We denote the branches $B^1, \ldots, B^t$. We consider the orientation of the branches such that o is always the upstream extremity of a branch. For a solution $C = \{c_1, \ldots, c_q\}$, if $o \in C$, then we fix $c_q = o$. We also denote C^i the subset of solution C located on B^i, with $c_{q(i)} \in C^i$ the nearest center to o on B^i (except o itself if $o \in C$). The vertices of C are numbered such that $C^i \setminus \{o\} = \{c_{q(i-1)+1}, \ldots, c_{q(i)}\}$, with $q(0) = 0$. Note that $c_{q(i)}$ can be the pending vertex of B^i.

We observe that the set $C \cup \{o\}$ induces a set of segments as defined in Section 5.2 (the only difference being that o is not necessarily included in C). The segments are numbered increasingly from each pending vertex to o, such that c_j is always the extremity of μ_j^C the more distant from o. We illustrate these notations on Figure 5.4. We represent a path without shelters with a hashed line, while a dotted line represents a path with or without shelters. In this example, B^1 is the path whose extremities are c_1 and o and $c_{q(1)} = c_3$.

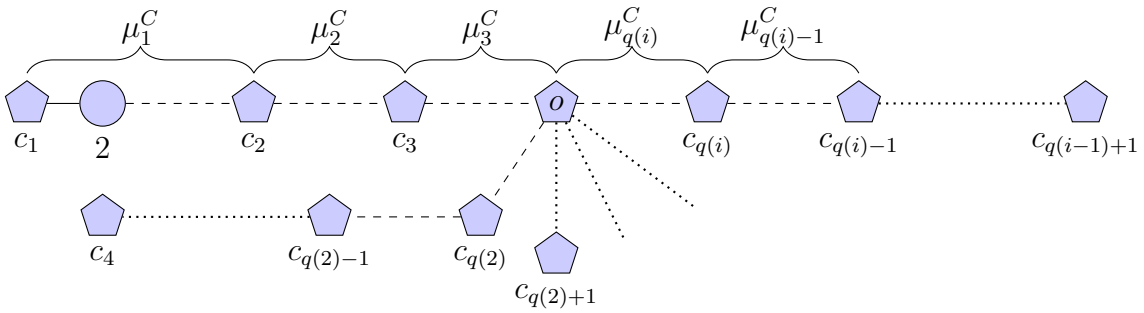


Figure 5.4: Example of notations on an extended star.

In addition, we can apply the notation introduced in Section 5.2 on μ_j^C. More precisely, we consider the configuration where $\beta_{q(i)}^C$ corresponds to the length of the

edge adjacent to o on $\mu^C_{q(i)}$. Note that these notations are independent of whether or not $o \in C$.

5.3.1 Computing the robust radius of a p-set

To compute the robust evacuation radius of a feasible solution C on S_t, we have to consider two cases: o is in solution C or not.

- If o is in the solution, then C induces $q-1$ segments, and the robust evacuation radius of C can be given like in Proposition 44, by:

$$\mathbf{R}(C) = \max_{1\leq i\leq q-1}\{\phi^C_i + \beta^C_i, \phi^C_i + \alpha^C_i\} \tag{5.2}$$

- If o is not in the solution, then $C \cup \{o\}$ induces q segments. The robust evacuation radius of C is given by

$$\mathbf{R}(C) = \max\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$$

where

$$\mathtt{a} = \max_{1\leq j\leq q, 1\leq i\leq t, j\neq q(i),}\{\phi^C_j + \beta^C_j, \phi^C_j + \alpha^C_j\} \tag{5.3a}$$

$$\mathtt{b} = \max_{1\leq i\leq t} \lambda^C_{q(i)} \tag{5.3b}$$

$$\mathtt{c} = \max_{1\leq i\leq t}\{\phi^C_{q(i)} + \beta^C_{q(i)} + \min_{1\leq j\leq t, j\neq i}(\alpha^C_{q(j)} + \phi^C_{q(j)} + \beta^C_{q(j)})\} \tag{5.3c}$$

  - (5.3a) gives the maximum evacuation distance of all vertices included on all segments μ^C_j such that $o \notin \mu^C_j$; it can be computed as in Proposition 44.
  - (5.3b) gives the evacuation distance of vertex o for scenario o.
  - (5.3c) gives the maximum evacuation distances of the vertices in $\mu^C_{q(i)}$. Note first that the downstream evacuation path of any vertex $x \in \mu^C_{q(i)}, x \notin C$ is contained in the downstream evacuation path of vertex $b_{q(i)}$ whose length is dominated by the evacuation distance of o (5.3b). On the other hand, the upstream evacuation path of x will cross o to reach a shelter on another branch. The upstream evacuation path of x is then contained in the upstream evacuation path of $a_{q(i)}$ whose length is given by $\phi^C_{q(i)} + \beta^C_{q(i)} + \min_{1\leq j\leq t, j\neq i}(\alpha^C_{q(j)} + \phi^C_{q(j)} + \beta^C_{q(j)})$).

For a given feasible solution C, we can compute α^C_i, ϕ^C_i and β^C_i for all values of i by scanning all edges once. Thus $\mathbf{R}(C)$ can be computed in $O(n)$.

5.3.2 Algorithm

Algorithm 4 Solving k-RCP- extended stars

Require: Edge-weighted extended star S_t rooted in o, integer k
Ensure: Returns C, k-RCP solution for S_t
1: $C^i \leftarrow$ Algorithm 3(B^i, k) , $\forall i \in \{1, \ldots, t\}$
2: $C \leftarrow \cup_{\forall i} C^i$
3: $C^* \leftarrow C \setminus \{o\}$
4: **if** $\mathbf{R}(C^*) \leq k$ **then**
5: Return C^*
6: **else**
7: Return C
8: **end if**

We propose Algorithm 4 to solve k-RCP on S_t. In the first step, using Algorithm 3, we solve k-RCP on each path B^i considered independently of S_t. C^i is the solution obtained on B^i. From these t solutions, we can compute in step 2 a unique solution $C = \cup_{i=1}^{t} C^i$ on S_t. By construction, C is a feasible solution for k-RCP on S_t. In addition, $o \in C$ as $o \in C^i$ for being the extremity of $B^i, i = 1, \ldots, t$. The remaining steps check whether there is a feasible solution of smaller size than C. We will see later that a smaller feasible solution than C exists if and only if $C^* = C \setminus \{o\}$ is a feasible solution on S_t. Thus, we compute $\mathbf{R}(C^*)$ using Equations 5.3 in step 4. If $\mathbf{R}(C^*) \leq k$, then C^* is feasible and and we return C^*. Otherwise C^* is not feasible and we return C.

We explain now the validity of our algorithm i.e., Algorithm 4 returns an optimal solution.

1. If $\mathbf{R}(C^*) > k$, then C^* is not feasible. If we want to construct a feasible solution of size $|C^*|$, then we need to relocate at least one shelter $c_{q(i)}$ for some i upstream on the branch B^i. However $c_{q(i)}$ can't be moved upstream without increasing $\lambda^{C^*}_{q(i)-1}$. Given properties P and P' (introduced in Section 5.2.3) on B^i, it is impossible to have a feasible solution of size $|C^*|$. As solution C is feasible by construction, C is an optimal solution and $|C| = |C^*| + 1$ is the minimum size of a solution.

2. If $\mathbf{R}(C^*) \leq k$, then C^* is feasible. If we want to construct a feasible solution of size $|C^*| - 1$, then we need to relocate at least one shelter $c^i_{q(i)}$ for a given i towards o. As for the previous case, this is impossible without violating the feasibility condition. Thus there is no feasible solution of size $|C^*| - 1$, and C^* is an optimal solution of minimum size.

Thus, we have proved the validity of the algorithm. Next we clarify its complexity. Let n_i be the number of vertices of $B^i \setminus o$, such that $1 + \sum_{1 \leq i \leq t} (n_i) = n$. Then Step 1 of Algorithm 4 is done in $\sum_{1 \leq i \leq t} O(n_i) = O(n)$, while computing $\mathbf{R}(C^*)$ is done in $O(n)$. Therefore Algorithm 4 is in $O(n)$.

In the next section, we consider the class of caterpillars.

5.4 Caterpillar

Consider an edge-weighted caterpillar G of n vertices, with $1, \ldots, \tilde{n}, \tilde{n} \leq n$ the vertices of the central path. We recall that $L \subset V$ is the set of pending vertices. In a caterpillar, we denote $L^c = L \setminus \{1, \tilde{n}\}$. We consider the central path of maximum length, such that the extremities of the central path are pending vertices. Considering an arbitrary orientation of the central path, we number first the vertices on the central path increasingly. The pending vertices adjacent to a vertex v are labeled v', v'', etc. This labeling is illustrated in Figure 5.5, for a caterpillar with $n = 16$ and $\tilde{n} = 10$.

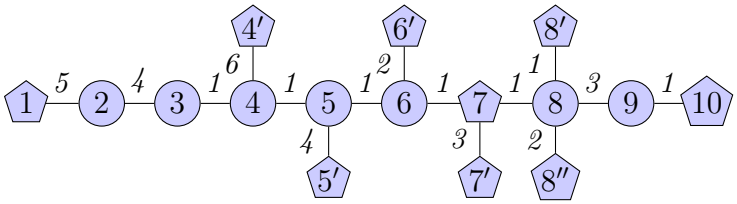


Figure 5.5: Example of the notation of a caterpillar.

On a path we have seen that, for a given feasible solution of `k-RCP`, every vertex not included in the solution has exactly two potential evacuation paths: an upstream evacuation path i.e., an evacuation path crossing vertices numbered in the increasing order, and a downstream evacuation path i.e., an evacuation path crossing vertices numbered in the decreasing order. Upstream or downstream describes the direction of an evacuation path depending on the direction they follow along the central path. We use the same notation for caterpillars. The evacuations paths can be characterized as upstream or downstream evacuation paths. When an evacuation path is a sole edge between a vertex of the central path and a pending vertex, its direction is said *horizontal*.

To build a feasible solution, we want to determine the set of vertices of the central path to be included in the solution in addition to the pending vertices. The caterpillar case has similarities with the path case, however we may encounter more complex situations compared to the path case: on a caterpillar and for a

given feasible solution, every vertex not included in the solution can have multiple potential evacuation paths, whereas on a path there are at most 2 evacuation paths. For example, we can see in Figure 5.5 a caterpillar with shelters located on the pending vertices and vertex 7. The edges are labeled by their length. We can observe that the nearest shelter to vertex 4 is located on vertex 7, at a distance of 3. However, in scenario 7, the shelter on 7 being unreachable, the evacuation path of vertex 4 goes through vertices 5 and 6 to shelter 6', at a distance 4. In scenario 6, the nearest shelter to vertex 4 is on vertex 5' at a distance of 5. In scenario 5, the nearest shelter to vertex 4 is on vertex 4' at a distance of 6. Finally, in scenario 4, the evacuation distance of vertex 4 corresponds to the longest evacuation path of vertex 4, that is its path to shelter 1, at a distance of 10. In summary, vertex 4 has 5 potential evacuation distances.

The aim of this section is to present an algorithm that solve `k-RCP` on a caterpillar in linear time. In addition, the idea introduced in this section will contribute to the design of an efficient algorithm on trees.

5.4.1 Computing the robust radius of a p-set

Let $C = \{c_1, \ldots, c_q\} \cup L^c$ be a feasible solution for `RpCP` on G, where $\{c_1, \ldots, c_q\}$ is the set of shelters located on the central path, ordered in increasing order ($c_1 = 1$ and $c_q = \tilde{n}$). For $1 \leq i \leq q-1$, we denote by T_i^C the subgraph induced by the segment of the central path delimited by the consecutive shelters c_i and c_{i+1} and the pending vertices adjacent to the internal vertices of this segment. T_i^C is a caterpillar and corresponds to the generalization of the concept of segment in paths to caterpillars. Hence in the following, we refer to T_i^C as a segment of G. Note that the pending vertices of c_i and c_{i+1} are not included in T_i^C. We call length of T_i^C the length of its central path.

For example in Figure 5.5, T_1^C is the subgraph of G induced by vertices $\{1, 2, 3, 4, 4', 5, 5', 6, 6', 7\}$, while T_2^C is the subgraph of G induced by vertices $\{7, 8, 8', 8'', 9, 10\}$. Note that vertex $7'$ is not contained in any of the two segments.

Let $\delta_d(v)$ be the maximum downstream evacuation distance of a vertex $v \in V$ and $\delta_u(v)$ be the maximum upstream evacuation distance of vertex v over all scenarios. Thanks to the acyclic nature of the graph, any evacuation path of a vertex $x \in T_i^C$ is always included in T_i^C. We define then $\delta_d(T_i^C)$ (resp. $\delta_u(T_i^C)$) as the maximum downstream (resp. upstream) evacuation distance of a vertex $v \in T_i^C$ i.e., $\delta_d(T_i^C) = \max_{v \in T_i^C} \delta_d(v)$ and $\delta_u(T_i^C) = \max_{v \in T_i} \delta_u(v)$.

Finally, for every pending vertex $y \in L^c$ such that x is adjacent to y, x has an horizontal evacuation distance from x to y induced by scenario x. This horizontal

evacuation distance is not recorded as an upstream or downstream evacuation distance. We denote by $z(x)$ the maximum horizontal evacuation distance of a vertex x and by $z(T_i^C)$ the maximum horizontal evacuation distance of an internal vertex $x \in T_i^C$ i.e., $z(T_i^C) = \max_{x \in T_i^C, x \notin C, y \in L^c \cap \Gamma(x)} l_{xy}$. If x has no horizontal neighbor, then $z(x) = 0$.

Then the robust evacuation radius of a solution C is given by

$$\mathbf{R}(C) = \max_{1 \leq i \leq q-1} \{\delta_d(T_i^C), \delta_u(T_i^C), z(T_i^C)\} \tag{5.4}$$

Note that, when building a solution C for `k-RCP`, it is sufficient to verify if the robust evacuation radius of the solution is less or equal to k. We use this equation to propose a first algorithm for `k-RCP` on a caterpillar.

5.4.2 A basic algorithm for `k-RCP`

We present in this subsection a generic method, Algorithm 5, to solve `k-RCP` for a caterpillar $G = (V, E)$. Algorithm 5 follows the same idea as Algorithm 3 for paths. We locate shelters on the central path greedily, such that the shelters are located as upstream as possible while ensuring a feasible solution.

In Algorithm 5, as any feasible solution contains at least the pending vertices, we focus on determining a subset of vertices C of the central path on which shelters must be located, such that the union of this subset and the pending vertices is an optimal solution of `k-RCP` for G. In the following, to simplify notations, a set $C \subseteq \{1, \ldots, \tilde{n}\}$ is said feasible for G if $C \cup L^c$ is a feasible solution for G. Besides C, a variable of Algorithm 5 is the set T, which corresponds to the most upstream segment induced by C. We initialize both variables C and T with $\{1, 2\}$ (steps 1 and 2). From steps 3 to 10, we iterate over the vertices 3 to $\tilde{n}$. At iteration v, we start by replacing $v - 1$ with v in C (step 4), and by adding v to T. At step 6, we verify whether $\delta_u(T)$ or $\delta_d(T)$ or $z(T)$ exceeds k. We do not specify a method to compute $\delta_u(T)$ and $\delta_d(T)$ for now. As for $z(T)$, we can easily compute it as we will see below. If $\delta_u(T)$ or $\delta_d(T)$ or $z(T)$ exceeds k, then we reinsert $v - 1$ in C (step 7) and T becomes the set $\{v - 1, v\}$ (step 8). At the end of the iteration, we return the set C (step 11).

Proposition 46. Algorithm 5 returns an optimal solution of `k-RCP` for a caterpillar and an integer k.

Proof. The proof is similar to the idea of the proof of Proposition 45. In the following, for $v \in \{2, \tilde{n}\}$, we refer to the state of variables C and T at iteration v

Algorithm 5 Solving `k-RCP` on a Caterpillar

Require: Edge-weighted caterpillar G
Ensure: Returns C, `k-RCP` solution for G
1: $C \leftarrow \{1, 2\}$
2: $T \leftarrow \{1, 2\}$
3: **for** $v \in [3, \ldots, \tilde{n}]$ **do**
4: $C \leftarrow C \setminus \{v-1\} \cup \{v\}$
5: $T \leftarrow T \cup \{v\}$ and add the pending vertices adjacent to $v-1$ to T
6: **if** $(\delta_d(T) > k)$ **or** $(\delta_u(T) > k)$ **or** $(z(T) > k)$ **then**
7: Add $\{v-1\}$ to C
8: $T \leftarrow \{v-1, v\}$
9: **end if**
10: **end for**
11: Return C

as $C(v)$ and $T(v)$ respectively. Note that we consider $v = 2$ at the initialization stage. In addition, we denote $G(v)$ the subgraph of G such that $\{1, \ldots, v\}$ is the central path of $G(v)$. Note that any pending vertex adjacent to v is not included in $G(v)$.

The statement to prove is $P(v)$: "$C(v)$ is an optimal solution for $G(v)$". A secondary statement useful for our proof is $P'(v)$: "The centers of $C(v)$ are located maximally upstream in $G(v)$". More precisely, $P'(v)$ states that, if any shelter is moved upstream in $G(v)$, then $\mathbf{R}(C(v)) > k$. We prove by induction that, $\forall v \in \{2, n\}$, $P(v)$ and $P'(v)$ are true.

Base case The base case is for $v = 2$. $C(2) = \{1, 2\}$ is trivially the optimal solution for $G(2)$. Thus $P(2)$ and $P'(2)$ are true.

Inductive step Assume $P(v)$ and $P'(v)$ are true at iteration v. We will prove that $P(v+1)$ and $P'(v+1)$ at iteration $v+1$. Let $C(v) = \{c_1, \ldots, c_{q^v}\}$, with $c_1 = 1$ and $c_{q^v} = v$. By step 4, $C(v+1) = C(v) \setminus \{v\} \cup \{v+1\}$. $C(v+1)$ includes vertices 1 and $v+1$, extremities of the central path of $G(v+1)$. We observe that the only segment that has been affected by the change of the solution is the most upstream segment $T(v+1)$. By $P(v)$, the evacuation distances of the other segments do not exceed k. Therefore, $C(v+1)$ is feasible if $\delta_u(T(v+1))$, $\delta_d(T(v+1))$ and $z(T(v+1)$ do not exceed k. In step 6, we compute $\delta_u(T(v+1))$, $\delta_d(T(v+1))$ and $z(T(v+1))$. As $z(T(v)) \leq k$ by $P(v)$, $z(T(v+1)) \leq k$ if and only if $z(v) \leq k$, otherwise $z(T(v+1)) > k$.

We identify two cases:

- if $C(v+1)$ is feasible, then, as $|C(v+1)| = |C(v)|$, $C(v+1)$ is of optimal size by$P(v)$. Hence, $P(v+1)$ is true. In addition, as the most upstream shelter has been relocated to the upstream extremity of the path, $P'(v+1)$ is true.

- if $C(v+1)$ is unfeasible, then we insert v back to $C(v+1)$ (step 7), such that $C(v+1) = C(v) \cup \{v+1\}$. By $P(v)$, $C(v)$ is feasible on $G(v)$. As $C(v) \subset C(v+1)$, $C(v+1)$ ensures evacuation distances less or equal to k on $G(v)$. Hence $C(v+1)$ is feasible. As the number of shelters of $C(v)$ were minimum ($P(v)$) and located as upstream as possible ($P'(v)$), no shelter in $G(v)$ can be relocated to an upstream vertex without breaking the feasibility condition. It follows that adding a shelter was necessary to satisfy the feasibility condition, thus $P(v+1)$ is true. By $P(v)$, the centers of $C(v)$ are located maximally upstream in $G(v)$. As $C(v) \subset C(v+1)$ and $(v+1) \in C(v+1)$, then the centers of $C(v+1)$ are located maximally upstream in $G(v+1)$. Thus $P'(v+1)$ is true.

Conclusion By induction, $P(v)$ and $P'(v)$ are true for all $v \in \{2, n\}$.

The algorithm ends after less than $\tilde{n}$ iterations and returns $C(\tilde{n})$ at the end of iteration $\tilde{n}$. As $G = G(\tilde{n})$, the returned solution is optimal for G. This concludes the proof. □

We can observe that the complexity of such an algorithm is $\tilde{n}$ times the complexity of computing $\delta_u(T(v))$ and $\delta_d(T(v))$, for $v = 1, \ldots, \tilde{n}$. If we want to determine these values by computing the evacuation distance of all the internal vertices, then the complexity depends on the number of vertices in T. In the worst case, $T(v)$ corresponds to $G(v)$ for each iteration $v = 3, \ldots, \tilde{n}$. Note that, if we consider that each internal vertex has at least one pending vertex, then $|L^c| \geq \tilde{n} - 2$ and $n = \tilde{n} + |L^c| \geq 2\tilde{n} - 2$. It follows that the complexity of Algorithm 5 is $O(1) + O(2) + \ldots + O(\tilde{n} - 2) = O(\tilde{n}^2) = O(n^2)$.

5.4.3 An improved algorithm for `k-RCP`

In this section, we introduce a new algorithm, Algorithm 6, which is an improvement of Algorithm 5. Algorithm 6 solves `k-RCP` on a caterpillar in $O(n)$ instead of $O(n^2)$ for Algorithm 5. More precisely, we show that, computing $\delta_d(T)$ and $\delta_u(T)$ are not required: we just need to check whether they exceed k. In Algorithm 6, changes compared to Algorithm 5 are colored in red. The red parts regarding the function z were actually already explained in the previous section, yet not explic-

itly introduced in the algorithm.

We introduce the parameter $h(v)$ for every vertex v of the central path. If v is adjacent to at least one pending vertex, then $h(v)$ is the minimum horizontal evacuation distance of v, otherwise we set $h(v) = k + 1$ (step 3). In addition, we introduce a new variable, φ defined below. As in the previous section, we refer to the state of variables C and T at iteration v as $C(v)$ and $T(v)$ respectively. We do the same for φ. Assume $C(v-1)$ is a feasible solution, we show that at the end of iteration v, $C(v)$ is a feasible solution.

We can observe that the only downstream evacuation path affected by the shelter relocation is the downstream evacuation path of $v-1$; the downstream evacuation path of the other vertices located downstream of v are unaffected. In addition, as a shelter is located on (v), $\delta_d(v) = 0$. Note that $\delta_d(T(v)) \neq \delta_d(v-1)$): for example, in Figure 5.5, $\delta_d(T(7))$ is given by $\delta_d(T(4)) = 10$, while $\delta_d(6) = 5$. However, if $\delta_d(T(v)) \leq k$, then $\delta_d(T(v))$ exceeds k if and only if $\delta_d(v-1)$ exceeds k. Consequently, it is sufficient to compute the downstream evacuation distance of $v-1$ to verify whether $\delta_d(T(v)) > k$. At initialization, that is for $v = 2$, we initialize $\delta_d(2) = 0$ at step 10. At iteration v, we compute $\delta_d(v-1)$ in $T(v)$ at step 14: the downstream evacuation path of $v-1$ is the sum of the distance from $v-1$ to $v-2$ and the minimum distance between $v-2$ and a shelter located downstream or horizontally.

The case $\delta_u(T(v))$ is more complex: when relocating the shelter from $v-1$ to v, the upstream evacuation path of a vertex x that ended on $v-1$ in $G(v-1)$, may be completely redirected to a pending vertex located between x and v in $G(v)$. We illustrate this situation with an example.

Example. Figure 5.6 shows the downstream portion of some edge-weighted caterpillar G for which a solution of `k-RCP` with $k = 5$ is build with Algorithm 5. The edges are labeled by their length. At iteration 7, the shelter located on 6 is relocated to vertex 7, as illustrated by the red arrow and the transparent shelter symbol on vertex 6. First consider the graph $G(6)$ at the end of iteration 6, when a shelter is located on 6. Note that vertex 6 is the nearest upstream shelter of vertices $2, 3, 4, 5$ and is located within a distance 5 of each of these vertices. Globally, $\delta_u(T(6)) = 4$. Therefore the upstream evacuation paths of $2, 3, 4, 5$ are ending in 6 at the end of iteration 6. Now we iterate on $G(7)$, and relocate the shelter from 6 to 7. We can observe that:

- for vertex 2, the distance to shelter 7 exceeds 5, however there is a shelter within a distance 5 in 3'. Then the upstream evacuation path of 2 ends in 3'.

- for vertex 3, the distance to shelter 7 exceeds 5, however there is a shelter within a distance 5 in 5'. Then the upstream evacuation path of 3 ends in 5'.
- for vertex 4, the distance to shelter 7 is 5, and there are shelters within a distance 5 in 5' and 6' too. As shelter 5' is the nearest at a distance 4, the upstream evacuation path of 4 ends in 5'.
- for vertex 5, the distance to shelter 7 is 4, and there is a shelter within a distance 5 in 6'. As shelter 7 is the nearest at a distance 4, the upstream evacuation path of 5 ends in 7.

Notice that, for vertices 2 and 3 who became out of range of shelter 7, their upstream evacuation paths have been rerouted to different shelters. In addition, vertex 4, despite being within radius 5 of shelter 7, has an upstream evacuation path ending in another shelter now. Finally, the upstream evacuation path of vertex 5 is the only upstream evacuation path that has been extended to end at shelter 7. In this case $\delta_u(T(7)) = 5$, induced by the upstream evacuation distance of vertices 2 and 3. Note that, if the length of edge $(5, 5')$ was 4 instead of 5, then vertex 3 would not have any upstream shelter within a distance 5.

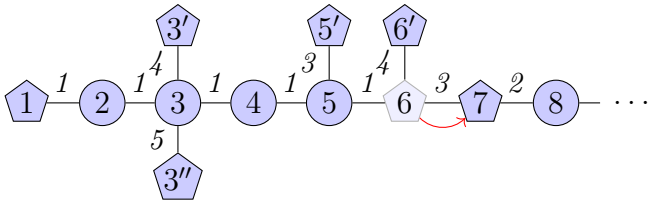


Figure 5.6: Applying Algorithm 5 on a caterpillar for `k-RCP` with $k = 5$, the red arc shows the relocation of the shelter from vertex 6 to vertex 7 at the beginning of iteration 7.

This example, illustrates that we can not easily compute $\delta_u(T(v))$ using $\delta_u(T(v-1))$, since some structural characteristics of $T(v-1)$ (the upstream evacuation paths) are not conserved in $T(v)$. ♣

Instead of computing all upstream evacuation distances, we show that $\delta_u(T(v)) > k$ if and only if the upstream evacuation distance of some critical vertices exceed k when the shelter is relocated from $v-1$ to v. At the end of iteration $v-1$ we observe that, for any $x \in T(v-1)$, there may be one or more shelters located upstream of x within a distance k. We define $X(v-1)$ the set of vertices such that, at the end of iteration $v-1$, the unique upstream shelter within a distance k of $x \in X(v-1)$ is $v-1$. Obviously, the upstream evacuation paths of the vertices of $X(v-1)$ are contained in the upstream evacuation path of the most downstream-located

vertex of $X(v-1)$, which we denote x^*. We define $\varphi(v-1) = \delta_u(x^*)$ at the end of iteration $T(v-1)$. Using $\varphi(v-1)$, we verify in step 15 if a pending vertex adjacent to $v-1$ is within a distance k of the vertices of $X(v-1)$. If so, then $x^* = v-1$ and $X(v) = \{v-1\}$. Consequently, $\varphi(v) = \varphi(v-1) = l_{v-1,v}$. Otherwise, x^* is unchanged and $X(v) = X(v-1) \cup \{v-1\}$. Consequently, $\varphi(v) = \varphi(v-1) + l_{v-1,v}$.

Algorithm 6 Solving `k-RCP`- Caterpillar

Require: Edge-weighted caterpillar $G = (V, E)$, integer k
Ensure: Give C, `k-RCP` solution for G

1: $T \leftarrow \{1, 2\}$
2: $C \leftarrow \{1, 2\}$
3: **for** $v \in \{2, \ldots, \tilde{n} - 1\}$ **do**
4: **if** $\Gamma(v) \cap L^c \neq \emptyset$ **then**
5: $z(v) \leftarrow \max_{x \in \Gamma(v) \cap L^c} \{l_{v,x}\}$; $h(v) \leftarrow \min_{x \in \Gamma(v) \cap L^c} \{l_{v,x}\}$
6: **else**
7: $z(v) \leftarrow 0$; $h(v) \leftarrow k + 1$
8: **end if**
9: **end for**
10: $\delta_d(2) \leftarrow 0$, $\varphi \leftarrow 0$
11: **for** $v \in \{3, \ldots, \tilde{n}\}$ **do**
12: $T \leftarrow T \cup \{v\}$ and add the pending vertices adjacent to $v-1$ to T
13: $C \leftarrow C \setminus \{v-1\} \cup \{v\}$
14: $\delta_d(v-1) \leftarrow l_{v-2,v-1} + \min\{\delta_d(v-2), h(v-2)\}$
15: **if** $\varphi + h(v-1) \leq k$ **then**
16: $\varphi \leftarrow l_{v-1,v}$
17: **else**
18: $\varphi \leftarrow \varphi + l_{v-1,v}$
19: **end if**
20: **if** $\delta_d(v-1) > k$ **or** $\varphi > k$ **or** $(z(v-1) > k)$ **then**
21: Add $\{v-1\}$ to C
22: $\delta_d(v-1) \leftarrow 0$, $\varphi \leftarrow 0$
23: **end if**
24: **end for**
25: Return C

In Algorithm 6, there are $\tilde{n}$iterations with $O(1)$ operations, therefore Algorithm 6 is in $O(n)$. It is an improvement over Algorithm 5 which is in $O(n^2)$.

In the next section, we consider the class of trees.

5.5 Trees

Let G be an edge-weighted tree. We root G on a non-pending vertex, and we label the vertices following a postorder traversal of the vertices of G (visit the children then the parent). Then, vertex 1 is a pending vertex and n is the root of G. We denote $G(v)$ the subgraph of G rooted in v. For a vertex v, we denote $p(v)$ its parent in G and $\phi(v) \subset V$ the set of its children. Note that, $\forall i \in \phi(v)$, we have $i < v$. An ancestor of a vertex is any other vertex on the path from the vertex to the root. A descendant is the inverse relationship of ancestor. An evacuation path of v is said upstream if it includes the parent of v, and downstream if it includes a child of v. If there is an upstream path between v and a vertex y, then we say y is located upstream of v. If y is a descendant of v, then we say y is located downstream of v. As in the caterpillar case, for a given feasible solution of `RpCP`, every vertex of G not included in the solution have multiple potential evacuation paths. In addition, any vertex $v \notin C$, such that v is of degree greater than 2 or $v = n$, will have several downstream evacuation paths in scenario v. In the following, the downstream (resp. upstream) evacuation distance of a vertex is the maximum length of one of its downstream (resp. upstream) evacuation paths for a given scenario. Finally, in this section, for a graph H and a set X, we will denote $X \cap H$ the set of vertices of X included in graph H.

5.5.1 Computing the robust radius of a p-set

Let $C = \{c_1, \ldots, c_q\} \cup L$ with $q = p - |L|$ be a feasible solution for `RpCP` on G, where $\{c_1, \ldots, c_q\}$ is the set of shelters located on the internal vertices of the tree. In addition, we denote $c_{q+1} = n$. If $n \in C$, then we assume $c_q = c_{q+1}$. We generalize the notion of segments from paths to trees. For $j \in \{1, \ldots, q+1\}$, the *segment-tree* $T(c_j)$ is the maximal subgraph of G rooted in c_j such that $T(c_j) \cap (C \cup \{c_{q+1}\})$ is the set including c_j and the set of pending vertices of $T(c_j)$. More precisely, for each pending vertex c_i in $T(c_j)$, c_i is a descendant of c_j and none of the internal vertices of the path between c_i and c_j is in C.

Thanks to the acyclic nature of the graph, the evacuation path of a vertex $x \in T(c_j)$ is always included in $T(c_j)$. We define then $\delta_d(T(c_j))$ (resp. $\delta_u(T(c_j))$) as the maximum downstream (resp. upstream) evacuation distance of a vertex $v \in T(c_j)$ over all scenarios, i.e., $\delta_d(T(c_j)) = \max_{v \in T(c_j)} \delta_d(v)$ and $\delta_u(T(c_j)) = \max_{v \in T(c_j)} \delta_u(v)$. Then the robust evacuation radius of a feasible solution C is:

$$\mathbf{R}(C) = \max_{1 \leq j \leq q+1} \{\delta_d(T(c_j)), \delta_u(T(c_j))\} \tag{5.5}$$

Next we propose an algorithm to iteratively build a solution by generalizing the approaches presented in the previous sections.

5.5.2 Algorithm for k-RCP

Algorithm 7 is a generalization of the k-RCP algorithms introduced in the previous sections: in the for-loop (steps 2-24), we perform a postorder traversal of the vertices of the tree, and we locate shelters such that no shelter can be relocated upstream without breaking the feasibility condition. When we visit v, we verify whether a shelter is needed on v or not. More precisely, we check whether having a shelter on $p(v)$ makes unnecessary to have a shelter on v. We denote, $\forall i \in \phi(v), G(v,i)$ the graph including of v and $G(i)$. Note that v is a pending vertex in $G(v,i)$.

We start with C an empty set. We label an iteration of the for-loop by the vertex under consideration. At iteration v, if v is a pending vertex (steps 3-5), then we add v and $p(v)$ to C and we initialize the parameters $\nabla_d(v)$ and $\varphi(v)$ to zero. We will define these parameters later on. At iteration v, if $v < n$ and $v \notin L$ (steps 6-16), then we add $p(v)$ to C and we remove v from C (step 7). Next, we check whether C induces a feasible solution on $G(p(v), v)$. If so, C is said feasible, otherwise C is not feasible.

We define $T(p(v), v) = T(p(v)) \cap G(p(v), v)$. We will see in Proposition 47 that C is feasible if and only if $\delta_d(v) \leq k$ and $\varphi(v) \leq k$ in $T(p(v), v)$. We recall that $\delta_d(v)$ is the maximum downstream evacuation distance of v. To compute $\delta_d(v)$ in $T(p(v), v)$, we use $\forall i \in \phi(i), \nabla_d(i)$ defined as the minimum downstream evacuation distance of vertex i in $T(v)$, which is unchanged in $T(p(v), v)$ (step 8). Similarly to the caterpillar case, in $T(p(v), v)$, $\varphi(v)$ is the maximum upstream evacuation distance of a vertex x^* in $T(p(v), v)$ with no upstream shelter other then $p(v)$ within a distance k.

To compute $\varphi(v)$, we use the intermediate subset $U(v) \subseteq \phi(v)$ such that $i \in U(v)$ if and only if there is a descendant x_i^* of i with no upstream shelter other then $p(v)$ within a distance k in $T(p(v), v)$(step 9). We consider that, if a min query is performed over an empty set, then it returns the value ∞; and if a max query is performed over an empty set, then it returns the value 0. If $U(i) \neq \emptyset$, then $\varphi(v)$ is given by the maximum distance $\forall i \in U(i)$ between x_i^* and $p(v)$, otherwise $x^* = v$ and $\varphi(v)$ is the upstream evacuation distance of v (step 10). Once $\delta_d(v)$ and $\varphi(v)$ are known, we can check the feasibility of C (step 11). If C is not feasible, then we re-include v in C (step 12), and we update $\nabla_d(v)$ and $\varphi(v)$ to zero in preparation of iteration $p(v)$ (step 13). If C is feasible, then $\varphi(v)$ does not change and $\nabla_d(v)$ must be updated (step 15).

Finally, at iteration n (steps 17-22), we remove n from C and we verify the

feasibility of C on G. We compute $\delta_d(n)$ based on $\nabla_d(i), \forall i \in \phi(n)$. In addition, we check $\forall i \in \phi(n)$ whether there is a descendant x_i^* of i whose upstream evacuation distance in $T(n)$ exceeds k (computed within step 20). If so, or if $\delta_d(n) > k$, then C is not feasible (see details in Proposition 47), and we re-include v to C. Finally, at step 25 we return solution C. Next, we prove that the solution returned is an optimal solution of `k-RCP` for the instance (G, k).

Proposition 47. Algorithm 7 returns an optimal solution of `k-RCP` for a tree and an integer k.

Proof. The proof is similar to the idea of the proof of Proposition 46. In the following, for $v \in \{1, n\}$, we refer to the state of variable C at iteration v as $C(v)$. We denote $Y(v) = C(v) \cap G(p(v), v)$. In the following, adding or removing a vertex to or from $Y(v)$ corresponds to adding or removing a vertex to or from $C(v)$. We will start by proving that the first $n-1$ iterations of the loop ensure that, at the beginning of iteration n and $\forall i \in \phi(n)$, $Y(i)$ is optimal on $G(n, i)$. In addition, the shelters of $Y(i)$ are located maximally upstream on $G(n, i)$ i.e., if any shelter of $Y(i)$ is moved upstream in $G(n, i)$, then the robust radius of $Y(i)$ exceeds k. Then, we show that the last iteration ensures that $C(n)$ is an optimal solution for G with the shelters located maximally upstream. $C(n)$ is the solution returned.

Let $P(v)$ be the statement: " $Y(v)$ is an optimal solution for $G(p(v), v)$"; and $P'(v)$ be the statement: "The centers of $Y(v)$ are located maximally upstream in $G(p(v), v)$". We prove by induction that, $\forall v \in \{1, n-1\}$, $P(v)$ and $P'(v)$ are true.

Base case The base case is for $v = 1$, where we consider $G(p(1), 1) = G[\{1, p(1)\}]$. Given step 4, we have $Y(1) = \{1, p(1)\}$. On $G(p(1), 1)$, $\mathbf{R}(Y(1)) = 0$ and $Y(1)$ is the only feasible solution, thus $P(1)$ and $P'(1)$ are true.

Inductive step Assume $P(v')$ and $P'(v')$ are true at iteration $v' < n-1$. We will prove that $P(v'+1)$ and $P'(v'+1)$ are true at iteration $v'+1$. Note that $v'+1$ is not necessary the same vertex as $p(v')$. In the following, to simplify the notations, we denote $v = v' + 1$.

If $v \in L$, then $G(p(v), v) = G[\{v, p(v)\}]$. Given step 4, we have $Y(v) = \{v, p(v)\}$. On $G(p(v), v)$, $\mathbf{R}(Y(v)) = 0$ and $Y(v)$ is the only feasible solution, thus $P(v)$ and $P'(v)$ are true.

If $v \notin L$, then $G(p(v), v) = p(v) \cup (\bigcup_{i \in \phi(v)} G(v, i))$. At the beginning of iteration v, $\forall i \in \phi(v), Y(i) \subseteq Y(v)$. By $P(i)$, $Y(i)$ is an optimal solution for $G(v, i)$. As v is a pending vertex in $G(v, i)$, then $v \in Y(i)$ and $v \in Y(v)$. At step 4, we

remove v from $Y(v)$ and we add $p(v)$ to $Y(v)$. We observe that $T(p(v), v)$ is the only segment-tree in $G(p(v), v)$ affected by this change. By $P(i), i \in \phi(v)$, the evacuation distances of the other segment-trees in $G(p(v), v)$ do not exceed k. Therefore, $Y(v)$ is feasible on $G(p(v), v)$ if and only if $\delta_d(T(p(v), v)) \leq k$ and $T(p(v), v) \leq k$. We check these as follows.

- Regarding $\delta_d(T(p(v), v))$, the downstream evacuation distances of the vertices in $T(p(v), v) \setminus \{v, p(v)\}$ are unchanged. By $P(i), i \in \phi(v)$, the downstream evacuation distances of these vertices do not exceed k. Thus, we have $\delta_d(T(p(v), v)) > k$ if and only if $\delta_d(v) > k$.

- As for $\delta_u(T(p(v), v))$, similarly to the caterpillar case, it exceeds k if and only if $\varphi(v)$ exceeds k.

We then have two cases:

- if $Y(v)$ is feasible, then $|Y(v)| = |\bigcup_{i \in \phi(v)} Y(i)|$. This situation is a bit similar to the extended star case, when we question the location of a shelter on the root. Given $\forall i \in \phi(v), P(i)$, at least $|Y(i)|$ shelters are needed to cover the descendants of i in $T(p(v), v)$. In addition, the shelter on $p(v)$ is mandatory ($p(v)$ is a pending vertex), and the shelters $Y(v) \cap Y(i)$ can not be moved upstream in $G(p(v), v)$ without breaking the feasibility condition ($P'(i)$). It follows that $P(v)$ and $P'(v)$ are true.

- if $Y(v)$ is not feasible, then $|\bigcup_{i \in \phi(v)} Y(i)|$ shelters are needed to cover v. Therefore, we re-insert v to $Y(v)$ such that $Y(v) = p(v) \cup \bigcup_{i \in \phi(v)} Y(i)$. By $P(i), \forall i \in \phi(v)$, $Y(i)$ is optimal on $G(v, i)$, thus $\bigcup_{i \in \phi(v)} Y(i)$ is optimal on $T(v)$ and by $P'(i)$ its shelters are located maximally upstream in $T(v)$. The shelter on $p(v)$ is mandatory, thus $Y(v)$ is optimal on $G(p(v), v)$. It follows that $P(v)$ and $P'(v)$ are true.

Conclusion By induction, $P(v)$ and $P'(v)$ are true for all $v \in \{1, n-1\}$.

Thus, at beginning of iteration n, $C(n) = \bigcup_{i \in \phi(n)} Y(i)$ is a feasible solution for G of size t. We remove the root from $C(n)$ and check the feasibility of the solution. As explained before, it is sufficient to check that the maximum evacuation distances in $T(n)$ does not exceed k. We compute the downstream evacuation distance of n, and the upstream evacuation distance of the vertices x_i^* in $G(n, i)$. We have two cases:

- $C(n)$ is not feasible. As seen in the extended star case, the shelters of $C(n)$ can not be relocated upstream without breaking the feasibility condition, no solution of size $t-1$ is feasible. Consequently, we re-include n in $C(n)$ and we return $C(n)$, an optimal solution for G.
- $C(n)$ is feasible. As the shelters of $C(n)$ can not be relocated upstream without breaking the feasibility condition, no solution of lower size is feasible. Consequently, we return $C(n)$, an optimal solution for G.

Hence the proof is complete. □

In Algorithm 7, at iteration v we perform at most $O(deg(v))$ operations, for example in steps 8 and 9, 15. Over n iterations, that is at most $O(m)$ iterations (we recall that m is the number of edges in the graph). As G is a tree, the complexity of Algorithm 7 is therefore in $O(n)$.

Algorithm 7 Solving `k-RCP` on a Tree

Require: Edge-weighted tree G, k
Ensure: Returns an optimal solution C for `k-RCP` on G

Root G on any non-pending vertex, and label the vertices following a post-order traversal. Initialize C as an empty set.
for $v = \{1, \ldots, n\}$ **do**
  if $(v \in L)$ **then**
    Add v and $p(v)$ to C.
    $\nabla_d(v) \leftarrow 0, \varphi(v) \leftarrow 0$
  else if $v < n$ **then**
    Add $p(v)$ to C and remove v from C.
    $\delta_d(v) \leftarrow \max_{i \in \phi(v)} \{l_{v,i} + \nabla_d(i)\}$
    $U(v) \leftarrow \{i \in \phi(v) : \varphi(i) + \min_{j \in \phi(v), j \neq i} \{l_{v,j} + \nabla_d(j)\} > k\}$
    $\varphi(v) \leftarrow \max_{i \in U(v)} \{\varphi(i)\} + l_{v,p(v)}$
    if $(\delta_d(v) > k)$ **or** $(\varphi(v) > k)$ **then** $\{v$ is necessary in $C\}$
      Add v to C.
      $\nabla_d(v) \leftarrow 0, \varphi(v) \leftarrow 0$
    else $\{v$ is not necessary in $C\}$
      $\nabla_d(v) \leftarrow \min_{i \in \phi(v)} \{l_{v,i} + \nabla_d(i)\}$
    end if
  else $\{v = n\}$
    Remove v from C.
    $\delta_d(n) \leftarrow \max_{i \in \phi(n)} \{l_{n,i} + \nabla_d(i)\}$
    if $\delta_d(n) > k$ **or** $(\max_{i \in \phi(n)} \{\varphi(i) + \min_{j \in \phi(n), j \neq i} \{l_{n,j} + \nabla_d(j)\} > k)$ **then**
      Add v to C.
    end if
  end if
end for
Return C

Conclusion

In this chapter we introduce an algorithm to solve RpCP using a binary search with k-RCP. Any algorithm for k-RCP can be integrated to our algorithm to solve RpCP. We propose several refined polynomial algorithms for RpCP on different classes of acyclic graph, which have structural properties that can be taken advantage of. In particular, as shelters must be located on pending vertices, they provide a backbone for the solution. Starting from these pending vertices, we propose a bottom up approach to locate incrementally shelters.

In the next chapter, we propose mathematical programming methods to compute exact solutions for both PpCP and RpCP on general graphs.

CHAPTER 6

MATHEMATICAL PROGRAMMING APPROACHES

Abstract

In this chapter we introduce integer programs for RpCP, k-RCP and PpCP. For real size instances, the sizes of the linear programs are huge so we propose a decomposition scheme to solve RpCP exactly, as well as different methods to compute lower and upper bounds. On the basis of experimental results, we propose a refined exact algorithm, which is the best current exact algorithm to solve RpCP. We test our algorithms on public p-Median instances and subgrid instances, the latter being more realist to our context.

Contents

6.1 Integer linear programming models for p-Center 125

6.1.1 Daskin's model 125

6.1.2 A set covering based approach 126

6.1.3 The second model 127

6.1.4 Size of the programs 129

6.2 Integer linear programming formulations for RpCP 129

6.2.1 MIP formulation for RpCP 129

6.2.2 IP formulation for RpCP 130

6.2.3 A set covering based algorithm 133

6.2.4 Additionnal inequalities 133
6.2.5 Size of the programs 134
6.3 Integer linear programming models for PpCP 134
6.3.1 MIP formulation for PpCP 135
6.3.2 IP formulation for PpCP 135
6.3.3 Size of the programs 137
6.4 An exact solution method for RpCP 139
6.4.1 Upper and lower bounds 139
6.4.2 Exact solution method 140
6.4.3 Experimental results 142

Introduction

In the two previous chapters, we have proposed polynomial methods to generate exact solutions or constant-approximation results for RpCP and PpCP. We now examine the case of general graphs, for which no polynomial method exists to solve both problems. In this chapter we propose mathematical programming methods to solve RpCP and PpCP. Our objective is to provide an exact algorithm for both problems.

We start by presenting in Section 6.1 some integer programming (IP) formulations for Min p-Center from existing exact solution methods for Min p-Center. On the basis of these models, we develop different IP and mixed-integer programming (MIP) based programs for RpCP in Section 6.2; and PpCP in Section 6.3. As these first programs appear to be quickly non-tractable (even to write the MIP instance with $n = 100$), we develop a MIP-based algorithm for RpCP in Section 6.4. We present extensive experimental results showing the tractability of our approach. Unfortunately, the designed exact algorithm can not be adapted to PpCP. A significant portion of the work presented in Section 6.4 has been published in [28].

In this chapter, with $G = (V, E), V = \{1, \ldots, n\}$ edge-weighted undirected graph, we introduce new matrixial notations associated to the edge-weighted graph $G = (V, E)$:

- (l_{ij}) is the matrix of edge length, with $l_{ii} = 0$ if $i \in V$ and $l_{ij} = +\infty$ if $(i, j) \notin E$. Note that (l_{ij}) is symmetrical.

- $SP = (d_{ij})$ is the matrix of the shortest path length between every couple of vertices, with $d_{ii} = 0$ and $d_{ij} = +\infty$ if there is no path between vertices i and j.

We recall that D is the finite list of distinct distances between pair of vertices in G, using the shortest distances. D is obtained by sorting in increasing order the T different finite values of the matrix SP: $D_{\min} = D_1 < D_2 < D_3 < \ldots < D_T = D_{\max}$. We consider this list as part of the instance, if necessary it can be computed in $O(n^2)$ from the matrix SP.

We extend some of these notations directly to operational graphs. For any scenario s, we denote:

- (l^s_{ij}) the matrix of edge length in G^s. Note that $\forall s \in V, s \neq j$, we have $l^s_{ij} = l_{ij}$. If $s = j$, then $l^s_{ij} = +\infty$. Therefore, (l^s_{ij}) is asymmetric.
- $SP^s = (d^s_{ij})$ the matrix of shortest path lengths from i to j in G^s.

In the following, for an IP program P, we denote by LP the program where the integer restriction on the variables are relaxed. The optimal value of the programs P and LP are denoted $v(P)$ and $v(LP)$, respectively.

6.1 Integer linear programming models for p-Center

The aim of this section is to present the main formulations for the `Min p-Center` problem and the auxiliary problem `Min r-Dominating Set`. In Section 6.1.1, we present the MIP formulation introduced by Daskin in [50]. In Section 6.1.2, we present an IP formulation for `Min r-Dominating Set`. In Section 6.1.3, we present a second IP formulation for `Min p-Center` introduced by Elloumi, Labbé and Pochet in [51]. In addition, we present a more recent formulation introduced by Calik and Tensel in [52], which is a variant of the second formulation. This last model will be the starting point for the IP formulations for `RpCP` and `PpCP` in the next sections.

6.1.1 Daskin's model

Daskin proposed in [50] the first MIP formulation for the `Min p-Center`, referred to as P_1^{det}. The decision variables of this first model are:

- for all $j = 1, \ldots, n$, y_j is a binary variable with $y_j = 1$ if a shelter is located on j and 0 otherwise,
- for all $i, j = 1, \ldots, n$, x_{ij} is a binary variable with $x_{ij} = 1$ if i is affected to j, and $x_{ij} = 0$ otherwise.

The formulation by Daskin can be stated as follows:

$$P_1^{det}\begin{cases} \min & r & & (1)\\ s.t. & \sum_{j=1}^{n} y_j = p & & (2)\\ & x_{ij} \le y_j & \forall i,j \in \{1,\dots,n\} & (3)\\ & \sum_{j=1}^{n} x_{ij} = 1 & \forall i \in \{1,\dots,n\} & (4)\\ & r \ge \sum_{j=1}^{n} d_{ij}x_{ij} & \forall i \in \{1,\dots,n\} & (5)\\ & y_j \in \{0,1\} & \forall j \in \{1,\dots,n\} & \\ & x_{ij} \in \{0,1\} & \forall i,j \in \{1,\dots,n\} & \end{cases}$$

Constraint (2) fixes the numbers of centers to be located. Constraints (3) ensure that vertex i is assigned to vertex j only if a shelter is located on j. Constraints (4) ensure that every vertex of the graph is assigned to exactly one center. Constraints (5) ensure that the distance from any vertex to its assigned center is less or equal than the radius r. The last constraints are the binary restrictions. Finally, the objective function (1) ensures the computation of a solution of minimum radius r.

6.1.2 A set covering based approach

Daskin also proposed in [50] a mathematical method with a set covering based algorithm. The radius value of the set covering problem is selected from an interval of real numbers between pre-determined lower and upper bounds. The approach is based on a binary search over the range of coverage distances to determine the smallest coverage distance that allows all vertices to be covered.

We denote SC_r, the following IP formulation for this set covering problem, taken for [51] :

$$SC_r\begin{cases} \min & \sum_{j=1}^{n} y_j & & (1)\\ s.t. & \sum_{j:d_{ij}\le r} y_j \ge 1 & \forall i \in \{1,\dots,n\} & (2)\\ & y_j \in \{0,1\} & \forall j \in \{1,\dots,n\} & \end{cases}$$

In SC_r the decision variables are the binary variables $y_j, j \in \{1,\dots,n\}$ similar to P_1^{det}. The objective function (1) minimizes the number of centers to be located while constraints (2) ensure that every vertex i is within a distance r of at least one center j.

Considering the linear relaxation LSC_r, we can obtain a lower bound for `Min p-Center` by solving a finite series of LSC_r programs. Indeed, if $v(LSC_r) > p$,

then $v(P_1^{det}) > r$. When r increases, $v(LSC_r)$ decreases and the best lower bound is the smallest value of r ensuring $v(LSC_r) \leq p$. We can then perform a binary search on D to determine such lower bound solving several LSC_r [50].

Actually, this approach has been described in Section 2.2.1.1 using `Min r-Dominating Set`: a binary search on D is performed to find the minimum radius r^* for which the optimal value of `Min r-Dominating Set` is less or equal to p. The resulting solution is an optimal solution for `Min p-Center`. `Min r-Dominating Set` is a particular case of `Min Set Cover`.

6.1.3 The second model

6.1.3.1 Original formulation

Elloumi, Labbé and Pochet [51] proposed a formulation P_2^{det} using the fact that the optimal value of `Min p-Center` is restricted to a finite set of distance values. The decision variables of the following model are:

- the binary variables $y_j, j \in \{1, \ldots, n\}$ introduced in P_1^{det}.
- the binary variables $z_t, t \in \{2, \ldots, T\}$ where $z_t = 0$ if and only if there is a solution such that all vertices are at a distance of at most D_{t-1} from a center of the solution. Then $z_t = 1$ if and only if there is at least one vertex with no center at a distance less or equal of D_{t-1}.

The previous definition involves that, if $z_t = 0$ then $z_{t+1} = z_{t+2} = \ldots = z_T = 0$, and the objective function is strictly lower than D_t. Similarly, $z_t = 1$ implies $z_{t-1} = \ldots = z_1 = 1$.

$$
P_2^{det} \begin{cases}
\min & D_1 + \sum_{t=2}^{T} (D_t - D_{t-1}) z_t & & & (1) \\
s.t. & \sum_{j=1}^{n} y_j \leq p & & & (2) \\
 & z_t + \sum_{j : d_{ij} \leq D_{t-1}} y_j \geq 1 & \forall i \in \{1, \ldots, n\}, & t \in \{2, \ldots, T\} & (3) \\
 & y_j \in \{0, 1\} & \forall j \in \{1, \ldots, n\} & & \\
 & z_t \in \{0, 1\} & \forall t \in \{2, \ldots, T\} & &
\end{cases}
$$

Constraint (2) limits the numbers of centers to be located. Constraints (3) are inspired by constraints (2) in SC_r: in an optimal solution for P_2^{det}, constraints (3) ensures that for a given t, $z_t = 0$, if and only if all clients can be served at a distance strictly lower than D_t. More precisely, if for a vertex i there is no center j within a distance D_{t-1}, then $z_t = 1$. It follows that, when $z_t = 1$ a distance

$D_t - D_{t-1}$ is added to the radius in the objective function (1). Without constraint (3), all variables z_t would equal to zero. The objective function (1) ensures that all vertices are covered by a center within a radius of the smallest possible value.

The authors proved in [51] that the optimal objective value of LP_2^{det} is greater or equal than the optimal objective value of LP_1^{det}. They also proved that a semi-relaxation of P_2^{det}, obtained by removing the binary restriction on the y variables, provides the best known lower bound for `Min p-Center`.

6.1.3.2 Variant formulation

In [52], Calik and Tansel introduce the formulation P^{det}, variant of the model proposed by Elloumi, Labbé and Pochet in [51]. The variables of the problem are:

- the binary variables $y_j, j \in \{1, \ldots, n\}$ similar to P_2^{det}.
- the binary variables $u_t, t \in \{1, \ldots, T\}$, where $u_t = 1$ if the value of the solution is equal to D_t and 0 otherwise.

$$
P^{det} \begin{cases}
\min & \displaystyle\sum_{t=1}^{T} D_t u_t & & (1^{det}) \\
s.t. & \displaystyle\sum_{j=1}^{n} y_j \leq p & & (2^{det}) \\
& \displaystyle\sum_{t=1}^{T} u_t = 1 & & (3^{det}) \\
& \displaystyle\sum_{j:d_{ij} \leq D_t} y_j \geq \sum_{q=1}^{t} u_q & \forall i \in \{1, \ldots, n\}, \forall t \in \{1, \ldots, T\} & (4^{det}) \\
& y_j \in \{0, 1\} & \forall j \in \{1, \ldots, n\} & \\
& u_t \in \{0, 1\} & \forall t \in \{1, \ldots, T\} &
\end{cases}
$$

Constraint (2^{det}) limits the number of shelters to be located. Constraint (3^{det}) ensures that exactly one variable u_t is equal to 1 and the corresponding D_t value is selected as the objective value according to the objective function (1^{det}). Constraints (4^{det}) ensure for each vertex i that at least one shelter is located at a distance less or equal than D_t. The authors tightened the constraints by taking into account that: if $u_t = 1$, then $\sum_{q=1}^{t} u_q = 1$. The semi relaxation of P^{det} obtained by removing the binary restriction on the y variables, provides the same tight lower bound as the semi relaxed P_2^{det}.

When lower and upper bounds are available,we can remove the solutions with such values and drop associated u_t variables from the model, thus decreasing the

size of the problem to be solved. Knowing a lower bound LB and an upper bound UB for the optimal value of P^{det}, we can set:

$$\begin{aligned} u_t &= 0 \quad ,\forall t : D_t < LB \\ u_t &= 0 \quad ,\forall t : D_t > UB \end{aligned}$$

Using this property, the algorithm developed by Calik and Tansel [52] solves their formulations for restricted sets of radius values iteratively to converge to an optimal solution.

6.1.4 Size of the programs

We present in Table 6.1 the number of variables for each formulation P_1^{det}, P_2^{det} and P^{det}, as well as the number of constraints generated. We recall that T is the size of the set D of candidate values for `Min p-Center` on G.

	P_1^{det}	P_2^{det}	P^{det}
Binary variables nb.	n^2+n	$n+T-1$	$n+T$
Real variables nb.	1	0	0
Constraints nb.	n^2+2n+1	$n(T-1)+2$	$nT+2$

Table 6.1: Number of variables and constraints for each presented formulation.

In particular, note that the number of binary variables of P^{det} is equal to $n+T$ and the number of constraints is $nT+2$. The size of this model depends on the number T of distinct shortest path lengths, which is in $O(n^2)$.

In the next section we extend the previous models to `RpCP`.

6.2 Integer linear programming formulations for RpCP

In this section we introduce three formulations for `RpCP`, as well as a formulation for the auxiliary problem `k-RCP`.

6.2.1 MIP formulation for RpCP

We extend formulation P_1^{det} to `RpCP` to propose formulation P_1^{rob}. We define the binary decision variable x_{ij}^s such that: if vertex i is assigned to a shelter located on vertex j in scenario s, then $x_{ij}^s = 1$, else $x_{ij}^s = 0$.

$$
P_1^{rob}\begin{cases}
\min & r & & (1)\\
s.t. & \sum_{j=1}^{n} y_j = p & & (2)\\
 & x_{ij}^s \le y_j & \forall i,j,s \in \{1,\dots,n\} & (3)\\
 & \sum_{j=1}^{n} x_{ij}^s = 1 & \forall i,s \in \{1,\dots,n\} & (4)\\
 & r \ge \sum_{j=1}^{n} d_{ij}^s x_{ij}^s & \forall s,i \in \{1,\dots,n\}, i \ne s & (5)\\
 & r \ge l_{sv}(1-y_s) + \sum_{j=1}^{n} d_{vj}^s x_{vj}^s & \forall s \in \{1,\dots,n\},\ \forall v \in \Gamma(s) & (6)\\
 & y_j \in \{0,1\} & \forall j \in \{1,\dots,n\} & \\
 & x_{ij}^s \in \{0,1\} & \forall i,j,s \in \{1,\dots,n\} &
\end{cases}
$$

The first 3 constraints have the same function as the first 3 constraints of formulation P_1^{det}. Constraint (2) fixes the number of centers. Constraints (3) ensure that vertex i is not assigned to a vertex j in scenario s unless j is a center. Constraints (4) ensure that every vertex is assigned to exactly one center for each scenario. Constraints (5) ensure that every vertex $i \neq s$ is within a distance r of at least one center in G^s for every scenario s. Constraints (6) are specific to RpCP and allow to model the chosen evacuation strategy:

- if $y_s = 1$, then a shelter is located on s and constraints (6) are relaxed;
- if $y_s = 0$, then no shelter is located on s and the set of constraints (6) on all neighbours of s ensure that the evacuation distance (worst case value) in scenario s is less than or equal to r

Finally the objective function (1) is to minimizes the robust evacuation radius.

6.2.2 IP formulation for RpCP

Based on formulation P_2^{det} and P^{det}, we propose two formulations for RpCP. We need first additional parameters. We have to replace D by D^{rob}, the list of distinct finite evacuation distance values in all G^s considering the evacuation strategy. The list D^{rob} is obtained by merging and ordering all the sets D^s of distinct finite evacuation distances between nodes in G^s, for $s \in V$. For each s, D^s is computed in two stages:

- step 1: initialize D^s with the different finite shortest path lengths SP^s for all $i \neq s$.

- step 2: $\forall j \neq s$ and $\forall v \in \Gamma(s)$ compute the distance $l_{sv} + d^s_{vj}$ from s to v to j and add it to D^s.

Finally, we merge all the lists D^s in one ordered set $D^{rob} = \{D^{rob}_1, \ldots, D^{rob}_{T^{rob}}\}$, with $D^{rob}_{\min} = D^{rob}_1 < D^{rob}_2 < D^{rob}_3 < \ldots < D^{rob}_{T^{rob}} = D^{rob}_{\max}$. In the next formulations, we use the y_j decision variables similar to P^{det}.

First formulation:

We introduce the binary decision variables z_t, inspired by P^{det}_2, such that $z_t = 0$ if and only if there is a solution such that for all scenarios, the evacuation distance of all vertices are less than or equal to D^{rob}_{t-1}. We propose then the following formulation:

$$
P^{rob}_2 \begin{cases}
\min \quad D^{rob}_1 + \sum_{t=2}^{T^{rob}} (D^{rob}_t - D^{rob}_{t-1}) z_t & & (1) \\
s.t. \quad \sum_{j=1}^{n} y_j \leq p & & (2) \\
\quad z_t + \sum_{j: d^s_{ij} < D^{rob}_t} y_j \geq 1 & \forall s, i \in \{1, \ldots, n\}, i \neq s & (3) \\
& \forall t \in \{2, \ldots, T^{rob}\} & \\
\quad z_t + \sum_{j: l_{sv} + d^s_{vj} < D^{rob}_t} y_j \geq 1 - y_s & \forall s \in \{1, \ldots, n\},\ \forall v \in \Gamma(s), & (4) \\
& \forall t \in \{2, \ldots, T^{rob}\} & \\
\quad y_j \in \{0, 1\} & \forall j \in \{1, \ldots, n\} & \\
\quad z_t \in \{0, 1\} & \forall t \in \{2, \ldots, T^{rob}\} &
\end{cases}
$$

Constraint (2) limits the number of centers. Constraints (3) ensure that, for a given t, $z_t = 0$ if and only if, in every scenario s, all vertices $i \neq s$ are within a distance less than D^{rob}_t of a center. Like constraints (6) for P^{rob}_1, constraints (4) are specific to RpCP and are necessary to compute the evacuation distance of the source vertex with the UP evacuation strategy. The objective function (1) ensures that the robust evacuation radius is the smallest one.

Second formulation:

We introduce the binary decision variables u_t, inspired by P^{det}, such that $u_t = 0$ if and only if, for any given scenario, all the vertices are at an evacuation distance of a center less than or equal to D^{rob}_t, in which case D^{rob}_t is the evacuation robust radius. We propose then the following formulation:

$$
P^{rob}\begin{cases}
\min & \displaystyle\sum_{t=1}^{T^{rob}} D_t^{rob} u_t & & (1^{rob}) \\
s.c. & \displaystyle\sum_{j=1}^{n} y_j \leq p & & (2^{rob}) \\
& \displaystyle\sum_{t=1}^{T^{rob}} u_t = 1 & & (3^{rob}) \\
& \displaystyle\sum_{j:d_{ij}^s \leq D_t^{rob}} y_j \geq \sum_{q=1}^{t} u_q & \forall s, i \in \{1,\ldots,n\}, i \neq s & \\
& & \forall t = 1,\ldots,T^{rob} & (4^{rob}) \\
& \displaystyle\sum_{j:l_{sv}+d_{vj}^s \leq D_t^{rob}} y_j \geq \sum_{q=1}^{t} u_q - y_s & \forall s = 1,\ldots,n, \forall v \in \Gamma(s), & \\
& & \forall t \in \{1,\ldots,T^{rob}\} & (5^{rob}) \\
& y_j \in \{0,1\} & \forall j \in \{1,\ldots,n\} & \\
& u_t \in \{0,1\} & \forall t \in \{1,\ldots,T^{rob}\} &
\end{cases}
$$

Constraint (2^{rob}) fixes the number of centers. Constraint (3^{rob}) are similar to constraints (3^{det}) with the only difference that T is replaced with T^{rob}. Constraints (4^{rob}) ensure that, for every scenario s, each vertex $i \neq s$ has at least one shelter located within a distance D_t. Like constraints (6) for P_1^{rob} and constraints (4) for P_2^{rob}, constraints (5^{rob}) are specific to RpCP and are necessary to compute the evacuation distance of the source vertex in this model.

Note that, in P^{det} the constraint corresponding to (2^{rob}) limits the number of centers rather than fixing it. As Daskin, in his formulation, fixes the number of centers, we conducted basic experiments to compare the efficiency of P^{rob} with the size of a solution fixed or restricted. The formulation P^{rob} was never outperformed by the formulation with the constraint of restricting the size.

Finally, note that the size of models P_2^{rob} and P^{rob} depends on the size of the list D^{rob}. Similarly to P^{det}, the size of P^{rob} can be reduced knowing a lower bound LB and an upper bound UB for $v(P^{rob})$ since some variables can be fixed as follows:

$$
\begin{aligned}
u_t &= 0 \quad ,\forall t : D_t^{rob} < LB \\
u_t &= 0 \quad ,\forall t : D_t^{rob} > UB
\end{aligned}
$$

6.2.3 A set covering based algorithm

It is also possible to extend the IP formulation SC_r to our robust model. We introduce the formulation SC_r^{rob}, extension of the formulation SC_r for RpCP. SC_r^{rob} is actually an IP formulation of the problem k-RCP. We recall that a binary search can be performed on k-RCP to find the minimum radius r^* for which the optimal value of k-RCP is less or equal to p. The corresponding solution is then an optimal solution for RpCP.

$$SC_r^{rob} \begin{cases} \min & \sum_{j=1}^{n} y_j & & (1) \\ s.c. & \sum_{j:d_{ij}^s \leq r} y_j \geq 1 & \forall s, i \in \{1,\dots,n\}, i \neq s & (2) \\ & \sum_{j:l_{sv}+d_{vj}^s \leq r} y_j \geq 1 - y_s & \forall s \in \{1,\dots,n\}, \forall v \in \Gamma(s) & (3) \\ & y_j \in \{0,1\} & \forall j \in \{1,\dots,n\} & \end{cases}$$

Constraints (2) ensure that every vertex i is within a radius of value r of a center in every scenario s and constraints (3) are necessary to compute the evacuation distance of the source vertex. Keeping in with the objective function of SC_r, SC_r^{rob} minimizes the number of centers to be located.

6.2.4 Additionnal inequalities

Given an instance (G, p) of RpCP, where G is an edge-weighted graph and $p \geq 2$, we have seen in Section 2.1.5, Proposition 3, that any feasible solutions must include at least one vertex in each minimal articulation component of G.

In our formulations, this property is ensured as for a scenario s, $d_{ij}^s = +\infty$ if there is no path between i and j in G_s. In practice, d_{ij}^s is set to a very high value, which means these cases are not discarded automatically as they could be. We can consolidate the previous property by adding constraints (6^{rob}). We recall the notation $\Upsilon(G)$ for the set of minimal articulation components of G.

$$\sum_{j \in A} y_j \geq 1 \quad \forall A \in \Upsilon(G) \qquad (6^{rob})$$

In some cases, for example when the graph is a forest, we can straightforward force pending vertices to be centers, rather than have redundant 6^{rob} constraints. Constraints (7^{rob}) are then sufficient.

$$y_j = 1 \quad \forall j : \quad deg(j) = 1 \quad (7^{rob})$$

6.2.5 Size of the programs

We present in Table 6.2 the number of variables for each formulation P_1^{rob}, P_2^{rob} and P^{rob}, and the number of constraints generated by each formulation, without inequalities (6^{rob}) and (7^{rob}). We recall that $m = |E|$.

	P_1^{rob}	P_2^{rob}	P^{rob}
Binary variables nb.	$n^3 + n$	$n + T^{rob} - 1$	$n + T^{rob}$
Real variables nb.	1	0	0
Constraints nb.	n^3+2n^2+2m+1	$(n^2 - n)(T^{rob} - 1) + 2m(T^{rob} - 1)$	$n^2T^{rob} + 2mT^{rob}$

Table 6.2: Number of variables and constraints for each presented formulation of `RpCP`.

In P_1^{rob} constraints (6) implies $deg(v)$ constraints for every vertex $v \in V$ in the graph. Then the number of constraints is the sum of the vertices degrees of the graph, which is equal to $2m$. Similarly, constraints (4) and (5^{rob}) implies the factor $2m$ for the number of constraints of P_2^{rob} and P^{rob} respectively.

Compared to the original formulations for `Min p-Center`, the above introduced formulations are clearly bigger: generally, the number of variables and of constraints are increased by a factor n. This is not surprising, as the uncertainty considered in `RpCP` forces us to take into consideration all n scenarios. Finally, we point out that half of the constraints for P_2^{rob} and P^{rob} are induced by constraints (4) and (5^{rob}). These constraints are specific to our evacuation strategy, particularly to compute the evacuation distance of the source vertex. The other half of the constraints are linked to the evacuation distances of the vertices distant of the fire.

In the next section we extend the previous models to `PpCP`.

6.3 Integer linear programming models for PpCP

In this section we propose three different formulations for `PpCP`, based on the formulation for `RpCP` presented in the previous section. We extend formulations P_1^{rob}, P_2^{rob} and P^{rob} to `PpCP`.

We recall that we assume an uniform probability distribution over the scenarios, more precisely the probability of scenario s is $\frac{1}{n}$.

6.3.1 MIP formulation for PpCP

We extend formulation P_1^{rob} to PpCP to propose formulation P_1^{pro}. We can use the same decision variables y_j and x_{ij}^s as for P_1^{rob}. In addition, we introduce the variables r^s associated with scenario $s \in V$.

$$
(P_1^{pro}) \begin{cases}
\min & \displaystyle\sum_{s=1}^{n} \frac{1}{n} r^s & & (1) \\
s.t. & \displaystyle\sum_{j=1}^{n} y_j = p & & (2) \\
& x_{ij}^s \leq y_j & \forall i,j,s \in \{1,\ldots,n\} & (3) \\
& \displaystyle\sum_{j=1}^{n} x_{ij}^s = 1 & \forall s,i \in \{1,\ldots,n\} & (4) \\
& r^s \geq \displaystyle\sum_{j=1}^{n} d_{ij}^s x_{ij}^s & \forall s,i \in \{1,\ldots,n\}, i \neq s & (5) \\
& r^s \geq l_{sv}(1-y_s) + \displaystyle\sum_{j=1}^{n} d_{vj}^s x_{vj}^s & \forall s \in \{1,\ldots,n\},\ \forall v \in \Gamma(s) & (6) \\
& y_j \in \{0,1\} & \forall j \in \{1,\ldots,n\} & \\
& x_{ij}^s \in \{0,1\} & \forall i,j,s \in \{1,\ldots,n\} &
\end{cases}
$$

The main difference in complexity between PpCP and RpCP is the need for PpCP to register the evacuation radius of every scenario s. This can be easily adapted on formulation P_1^{rob} to generate formulation P_1^{pro}. This is done by modifying constraints (5) and (6) from P_1^{rob} such that the evacuation radius r^s for every scenario s is computed. The objective function (1) ensures then the minimization of the expected evacuation radius over all scenarios. The other constraints remain the same.

6.3.2 IP formulation for PpCP

We propose two formulations for PpCP adapted from P_2^{rob} and P^{rob}. Instead of using D^{rob} solely, we will need to use the n lists D^s, introduced in Section 6.2.2. We consider D^s ordered such that the elements of D^s are denoted by $D^s_{\min} = D^s_1 < D^s_2 < D^s_3 < \ldots < D^s_{T^s} = D^s_{\max}$.

First formulation: We introduce the binary decision variable z_t^s: $z_t^s = 0$ if and only if there is a solution such that, for scenario s, all vertices are within a radius of value D^s_{t-1} of a center. Formulation P_2^{rob} can be adapted into formulation P_2^{pro}:

$$
(P_2^{pro})\begin{cases}
\min & \displaystyle\sum_{s=1}^{n}\frac{1}{n}*\left(D_1^s+\sum_{t=2}^{T^s}(D_t^s-D_{t-1}^s)z_t^s\right) & & (1)\\
s.c. & \displaystyle\sum_{j=1}^{n} y_j \le p & & (2)\\
& z_t^s+\displaystyle\sum_{j:d_{ij}^s<D_t^s} y_j \ge 1 & \forall s,i\in\{1,\dots,n\} & (3)\\
& & \forall t=2,\dots,T^s & \\
& z_t^s+\displaystyle\sum_{j:l_{sv}+d_{vj}^s<D_t^s} y_j \ge 1-y_s & \forall s\in\{1,\dots,n\},\forall v\in\Gamma(s), & (3)\\
& & \forall t=2,\dots,T^s+1 & \\
& y_j\in\{0,1\} & \forall j\in\{1,\dots,n\} & \\
& z_t^s\in\{0,1\} & \forall s\in\{2,\dots,T^s+1\} &
\end{cases}
$$

Constraints (3) and (4) are the extensions of the corresponding constraints in P_2^{rob} for PpCP. The objective value (1) is also transformed in order to compute the expected value of the evacuation radius over all scenarios, and ensure that it is the smallest possible.

Second formulation: Finally we adapt formulation P^{rob} for PpCP with P^{pro}. We introduce the variables u_t^s with $u_t^s = 1$ if in scenario s all vertices are within a radius of value D_t^s from a center, in which case $r^s = D_t^s$ with $s = 1,\dots,n$ and $t = 1,\dots,n$.

$$
(P^{pro})\begin{cases}
\min & \displaystyle\sum_{s=1}^{n}\frac{1}{n}*\sum_{t=1}^{T^s} D_t^s u_t^s & & (1)\\
s.c. & \displaystyle\sum_{j=1}^{n} y_j = p & & (2)\\
& \displaystyle\sum_{t=1}^{T^s} u_t^s = 1 & \forall s\in\{1,\dots,n\} & (3)\\
& \displaystyle\sum_{j:d_{ij}^s\le D_t^s} y_j \ge \sum_{q=1}^{t} u_q & \forall s,i\in\{1,\dots,n\}, i\neq s & \\
& & \forall t\in\{1,\dots,T^s\} & (4)\\
& \displaystyle\sum_{j:l_{sv}+d_{vj}^s\le D_t^s} y_j \ge \sum_{q=1}^{t} u_q - y_s & \forall s\in\{1,\dots,n\},\forall v\in\Gamma(s), & \\
& & \forall t\in\{1,\dots,T^s\} & (5)\\
& y_j\in\{0,1\} & \forall j\in\{1,\dots,n\} & \\
& u_t^s\in\{0,1\} & \forall s\in\{1,\dots,n\}, t\in\{1,\dots,T^s\} &
\end{cases}
$$

Constraints (3) ensures that for each scenario s, exactly one variable $u_t^s, t \in \{1, \ldots, T^s\}$ will be equal to 1, and the corresponding D_t^s value is the evacuation radius of the solution for scenario s. Constraints (4) ensure that, for every scenario s, each vertex $i \neq s$ is within a distance D_t^s of a center. Like constraints (5^{rob}) for the P_3^{rob}, constraints (5) are specific to the UP evacuation strategy and are necessary to compute the evacuation distance of the source vertex.

6.3.3 Size of the programs

We present in Table 6.3 the number of variables for each formulation P_1^{pro}, P_2^{pro} and P^{pro}, and the number of constraints generated by each formulation.

	P_1^{pro}	P_2^{pro}	P^{pro}
Binary variables nb.	$n^3 + n$	$n + \sum_{s=1}^{n}(T^s - 1)$	$n + \sum_{s=1}^{n} T^s$
Real variables nb.	1	0	0
Constraints nb.	$n^3 + 2n^2 + 2m + 1$	$(n^2 - n)\sum_{s=1}^{n}(T^s - 1) + 2m\sum_{s=1}^{n}(T^s - 1)$	$n^2\sum_{s=1}^{n} T^s + 2m\sum_{s=1}^{n} T^s$

Table 6.3: Number of variables and constraints for each formulation introduced for PpCP.

Compared the size of formulations P_2^{rob} and P^{rob} with formulations P_2^{pro} and P^{pro}, we observe that the probabilistic approach induces a quadratic increase in the number of constraints. This is not surprising, as the probabilistic approach considered in PpCP compels us to compute the evacuation radius of the solution for every scenario. The size of formulation P_1^{pro} on the contrary remains constant in comparison with P_1^{rob}.

Note that the inequalities introduced in Section 6.2.4 relative to feasibility conditions, are also valid for PpCP and for formulations P_1^{pro}, P_2^{pro} and P^{pro}.

In addition, we underline that, unlike for RpCP, it is not possible to extend the set covering problem SC_r to PpCP: for a given expected evacuation radius, there are numerous possible $r_1, \ldots, r^s, \ldots, r_n$ distribution values.

Our first experiments with the different formulations for RpCP and PpCP were quickly limited with the size of the instance. Starting from 30 vertices, trying to generate the instance on a computer with 16Gb of RAM would often freeze the computer. Hence we decided to develop a specific algorithm for RpCP. For a same graph, the size of our RpCP formulations is smaller than the size of our PpCP formulations. In addition, we have the auxiliary program SC_r^{rob} to help us address the RpCP problem. As a basis for our algorithm, we performed nonetheless experiments on small instances to identify the best performing RpCP formulation. P^{rob} appeared to be most efficient.

In the next section, a new exact algorithm is defined and extensive experimental results show the tractability of our approach.

6.4 An exact solution method for RpCP

In this section, we present an exact solution method for RpCP based on formulation P^{rob} introduced previously. The size of the model P^{rob} depends on the size of the list D^{rob} leading to huge integer linear programs: the first computational tests outlined that D^{rob} can be of size bigger than n and have revealed that the generation of the constraints (4^{rob}) and (5^{rob}) may take a substantial time.

Consequently, in order to obtain optimal solutions we have to reduce its size (fixing some variables) and to define specific exact algorithm based on a generalization of a binary search. To this end, we propose in Section 6.4.1 several methods to determine tight upper and lower bounds. In Section 6.4.2, we present an algorithm for RpCP based on a generalization of binary search. Finally in Section 6.4.3 we present the experimental results of our algorithm on different data sets.

6.4.1 Upper and lower bounds

We propose four different methods to compute upper and lower bounds for RpCP:

- The first method uses an optimal solution of P^{det} with the algorithm proposed in [51]. Obviously, the value of an optimal solution of RpCP can not be less than the optimal value of P^{det}. We denote $LB^1 = v(P^{det})$. When this solution is feasible for RpCP, its value gives an upper bound UB^1 for RpCP.

- The second method allow us to compute a lower bound for RpCP based on SC_r^{rob}, presented in Section 6.2.3. A binary search can be performed on D^{rob} to find the minimum radius r^* for which $v(LSC_{r^*}^{rob}) \leq p$. A lower bound for P^{rob} is then r^*, denoted by LB^2.

- In a third method, we randomly construct solutions and compute their value for RpCP. The lowest obtained value represents a second upper bound UB^2.

- The fourth method consists in considering P^{rob} without the constraints (4^{rob}). The obtained model, denoted by RP^{rob}, corresponds to the problem where only the evacuation distance of the node s is taken into account for scenario s. It reduces the number of constraints by n^2T^{rob}. The value of the obtained solution is a lower bound LB^3 for RpCP. Similarly to the first method, if RP^{rob} has an optimal solution, which is feasible for RpCP, it gives a third upper bound UB^3 for RpCP.

In our preliminary experiments, despite using these bounds for P^{rob}, the number of constraints and variables were still too high in order to solve exactly the problem, more precisely even to write the LP instance. For example, for an instance with

100 nodes, 200 edges and $T^{rob} = 400$, the number of constraints exceeds 4 millions. It took us more than 9 hours and 120 gigabytes of memory usage to obtain the optimal solution.

Thus we propose a general scheme using a generalization of binary search algorithm. As we will see in the experimental results section, the same instance using our algorithms can be solved in less than 40 seconds.

6.4.2 Exact solution method

Algorithm 8 σ-quantile search

Require: $P(D), LB, UB, \sigma \in \mathbb{N}, \sigma \geq 3$
Ensure: Returns the optimal value and an optimal solution to $P(D)$
while $UB \neq LB$ **do**
  $D' = \{D_{k_1}, \ldots, D_{k_\sigma}\} \leftarrow Fkernel(D, LB, UB, \sigma)$
  Solve $P(D')$
  Set $q \in \mathbb{N}^+$ such that $D_{k_q} \leftarrow$ Optimal value of $P(D')$
  $sol \leftarrow$ Optimal solution of $P(D')$
  if $q = 1$ **then**
    $UB \leftarrow LB$
  else
    $UB \leftarrow D_{k_q}$
    $LB \leftarrow D_{k_{q-1}}$
  end if
end while
Return LB and sol

Algorithm 9 $Fkernel$

Require: $D = \{D_1, D_2, \ldots, D_T\}, LB, UB, \sigma \in \mathbb{N}, \sigma \geq 3$
Ensure: Returns a subset of D
Find $k_1 \in \{1, \ldots, T\}$ such that $D_{k_1} = LB$
Find $k_\sigma \in \{1, \ldots, T\}$ such that $D_{k_\sigma} = UB$
$step \leftarrow \lfloor (k_1 + k_\sigma)/(\sigma - 1) \rfloor$
for $i \leftarrow 2$ **to** $\sigma - 1$ **do**
  $k_i \leftarrow k_{i-1} + step$
end for
Return $\{D_{k_1}, \ldots, D_{k_i}, \ldots, D_{k_\sigma}\}$

Consider $P(D)$ a linear programming formulation whose objective value (to be minimized) takes a value from an ordered set $D = \{D_1, D_2, \ldots, D_T\}$. Denote LB

and UB two initial lower and upper bounds for $v(P(D))$. A *σ-quantile search*, presented in Algorithm 8, can be used to solve $P(D)$ by solving at most $\lceil \log_\sigma(T+1) \rceil$ instances of $P(D')$ with $D' \subseteq D$. While $UB \neq LB$, we repeat the following steps:

- First we compute a restricted set $D' = \{D_{k_1}, \ldots, D_{k_\sigma}\} \subseteq D$ using function $Fkernel$ (Algorithm 9): we delete all values in D that are less than LB or greater than UB, then $D_{k_1} = LB$, $D_{k_\sigma} = UB$ and the intermediate values correspond to a $(\sigma - 1)$-quantile. So, between every two consecutive values of D' there are roughly the same number of values of D.

- Next we solve $P(D')$. Let $v(P(D')) = D_{k_q}$, then D_{k_q} is an upper bound for $v(P(D))$ as $D_{k_q} \in D$. In addition, there is no feasible solution in $P(D')$ with value $D_{k_{q-1}}$ and equivalently in $P(D)$, so $D_{k_{q-1}}$ is a lower bound for $v(P(D))$. Note that if $D_{k_q} = D_{k_1}$, then D_{k_1} is the optimal solution for $P(D)$.

- Finally, $LB = D_{k_{q-1}}$ and $UB = D_{k_q}$.

Note that for $\sigma = 3$, the σ-quantile search is actually a binary search.

Our initial exact algorithm to solve P^{rob} is presented in Algorithm 10. In a first step, all lower and upper bounds are computed and in a second step, a σ-quantile search is performed.

Algorithm 10 Exact Algorithm

Require: $G = (V, E), x, \sigma_1, \sigma_2$
Ensure: Returns the optimal value and an optimal solution to P^{rob}
1: Generate D^{rob}
2: Solve P^{det} and generate LB^1 and UB^1
3: Compute LB^2
4: Compute UB^2 by generating x random solutions
5: $LB^3, sol \leftarrow \sigma$ - quantile search$(RP^{rob}(D^{rob}), \max\{LB^1, LB^2\}, \min\{UB^1, UB^2\}, \sigma_1)$
6: $UB^3 \leftarrow$ Value of sol for P^{rob}
7: $UB \leftarrow \min\{UB^1, UB^2, UB^3\}$ and $LB \leftarrow \max\{LB^1, LB^2, LB^3\}$
8: $optValue, optSolution \leftarrow \sigma$-quantile search$(P^{rob}(D^{rob}), LB, UB, \sigma_2)$
9: Return $optValue$

In the next section, we evaluate the computational efficiency of the proposed algorithm.

6.4.3 Experimental results

We implement the Exact Algorithm in Python 3.7 for two sets of instances: p-median instances from OR-Library [86] and subgrids randomly generated. We generate the distance matrices $SP = (d_{ij})$ and $SP^s = (d^s_{ij})$ for all scenario $s \in V$ using networkx library 2.3. We execute our experiments on a server with 254Gb of RAM and 14 Intel Core (Haswell; no TSX) Processor at 2.3 Ghz. Mathematical programs are solved with CPLEX 12.9 (with MIPEmphasis option set to 0).

6.4.3.1 Experimental results on OR-Library Instances

The input data used for the computations are the 40 instances of the p-Median problem from the OR-Library ([86]) which are used also for solving the p-center problem ([51, 52]). n varies between 100 and 900 nodes and p varies between 5 and $\lfloor n/3 \rfloor$. In the following we focus on the instances which could be solved within 5 hours.

Table 6.4 contains the value of the upper and lower bounds computed for the instances ordered by the number of their nodes and the value of p. We mark with an $*$ the values of the bounds that are equal to optimal value. It stands out that on all the instances considered, $LB^3 = UB^3 = v(P^{rob})$. The equality between UB^3 and LB^3 is not mandatory since only evacuation paths of a subset of nodes are considered in LB^3. In fact, we record some instances in which $LB^3 < v(P^{rob})$ (see in Figure 6.2) and, in these cases, the computation time is more important. The equality between LB^3 and UB^3 means that the evacuation distance of node s induces the radius of an optimal solution in scenario s. It may be due to the fact that OR-Library instances are considerably sparse.

Concerning lower bounds, LB^2 is also a tight lower bound very close and often equal to $v(P^{rob})$, while LB^1 is the worst lower bound. Concerning upper bounds, UB^1 is globally better than UB^2.

The Exact Algorithm computes all pairs of lower and upper bounds. In order to better understand the trade-off between bounds quality and computational time, we compare the processing time of three variants of the Exact Algorithm:

- EA1 is a version of the Exact Algorithm in which only LB^1 and UB^1 are computed, its processing time is TEA1. This variant is mainly based on the resolution of the deterministic p-Center problem.
- EA2 is a version of the Exact Algorithm in which only LB^2 and UB^2 are computed, its processing time is TEA2. This variant is adapted from the Daskin's algorithm for the `RpCP`.

Instance	n	|E|	p	OPT	LB1	UB1	LB2	UB2	LB3	UB3
pmed1	100	200	5	222	127	251	221	263	222*	222*
pmed2	100	200	10	194	98	229	192	251	194*	194*
pmed3	100	200	10	191	93	226	186	240	191*	191*
pmed4	100	200	20	157	74	184	156	218	157*	157*
pmed5	100	200	33	115	48	144	115*	180	115*	115*
pmed6	200	800	5	180	84	208	180*	205	180*	180*
pmed7	200	800	10	156	64	163	155	180	156*	156*
pmed8	200	800	20	143	55	153	143*	188	143*	143*
pmed9	200	800	40	124	37	136	124*	164	124*	124*
pmed10	200	800	67	100	20	118	100*	136	100*	100*
pmed11	300	1800	5	153	59	157	153*	169	153*	153*
pmed12	300	1800	10	145	51	150	145*	171	145*	145*
pmed13	300	1800	30	129	36	136	128	154	129*	129*
pmed14	300	1800	60	116	26	125	115	145	116*	116*
pmed15	300	1800	100	105	18	118	105*	133	105*	105*
pmed16	400	3200	5	143	47	147	143*	153	143*	143*
pmed17	400	3200	10	136	39	139	136*	150	136*	136*
pmed18	400	3200	40	122	28	127	122*	144	122*	122*
pmed19	400	3200	80	112	18	119	111	132	112*	112*
pmed20	400	3200	133	103	13	113	103*	130	103*	103*
pmed21	500	5000	5	137	40	139	137*	147	137*	137*
pmed22	500	5000	10	133	38	137	133*	147	133*	133*
pmed23	500	5000	50	118	22	122	118*	135	118*	118*
pmed24	500	5000	100	110	15	115	110*	132	110*	110*
pmed25	500	5000	167	103	11	113	103*	131	103*	103*
pmed26	600	7200	5	134	38	137	134*	147	134*	134*
pmed27	600	7200	10	128	32	132	128*	139	128*	128*
pmed28	600	7200	60	114	18	118	114*	134	114*	114*
pmed29	600	7200	120	108	13	113	108*	130	108*	108*
pmed30	600	7200	200	103	9	109	103*	124	103*	103*
pmed31	700	9800	5	128	30	136	124	135	128*	128*
pmed32	700	9800	10	127	29	128	123	164	127*	127*
pmed33	700	9800	70	113	15	119	110	126	113*	113*
pmed34	700	9800	140	107	11	111	107*	125	107*	107*
pmed35	800	12800	5	128	30	130	-	136	128*	128*
pmed36	800	12800	10	125	27	127	-	134	125*	125*
pmed37	800	12800	80	112	15	121	-	130	112*	112*
pmed38	900	16200	5	127	29	129	-	138	127*	127*
pmed39	900	16200	10	122	23	127	-	174	122*	122*
pmed40	900	16200	90	111	13	113	-	123	111*	111*

Table 6.4: Optimal solution values and bound values for OR-Library instances

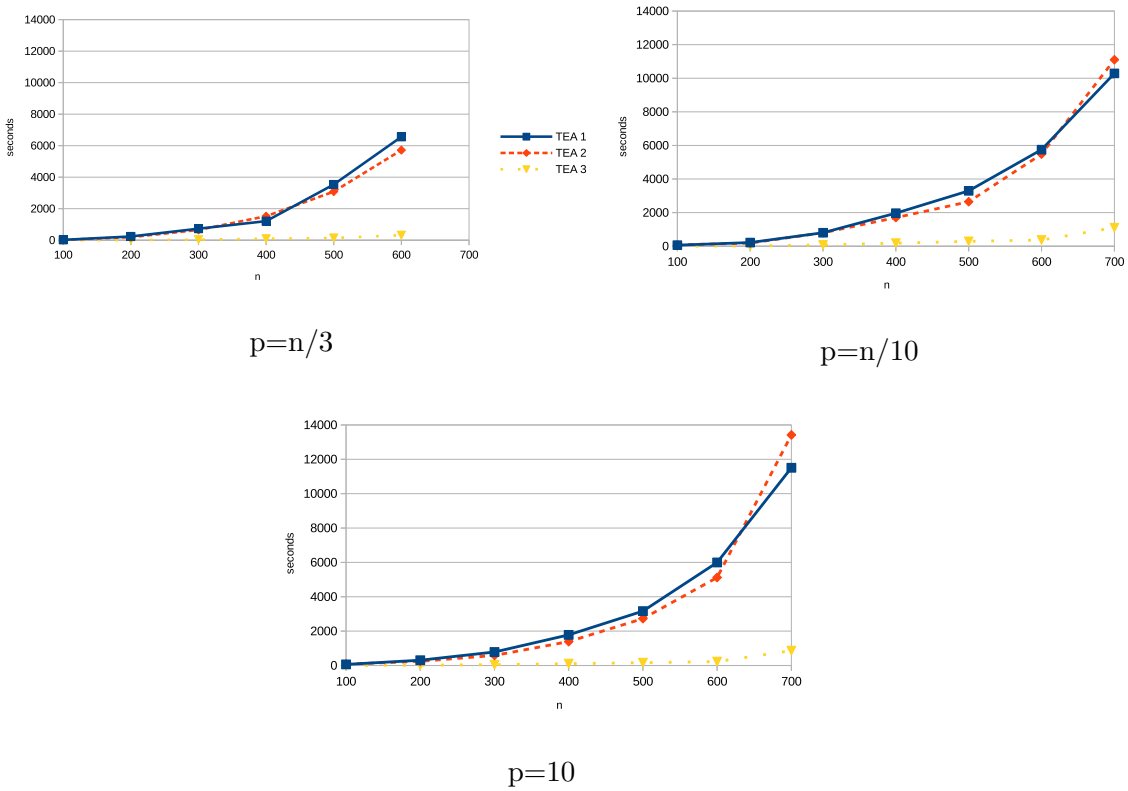


Figure 6.1: Processing time of each variant of the Exact Algorithm on some OR-Library instances

- EA3 is a version of the Exact Algorithm in which only LB^3 and UB^3 are computed, its processing time is TEA3. This variant is a new one specific to `RpCP` and its evacuation strategy.

For pmed35 to pmed40, TEA1 and TEA2 exceed five hours: only EA3 can exactly solve all instances in less than 5 hours.

In Figure 6.1, we represent the processing times for instances from pmed1 to pmed34 for three different values of p: $p = n/3$, $p = n/10$ and $p = 10$. For each p, three curves represent the processing times TEA1, TEA2 and TEA3, in function of n. We observe that our dedicated algorithm EA3 performs faster than EA1 and EA2. The reason is twofold: in EA1, the poor quality of the lower bound LB^1 (see Table 6.4) increases the number of iterations in step 8 of the Exact Algorithm. Conversely, in EA2 the quality of the lower bound LB^2 highly decreases the number of iterations in step 8, however computing LB^2 is very time consuming. More precisely, the generation of the input CPLEX instance for SC_r^{rob}

requires a large amount of time (at least 90% of the total computing time for generating LB^2). On the other hand, LB^3 and UB^3 can be computed much faster while providing the best quality of bounds. Thus, EA3 is clearly the most efficient of the three algorithms.

Once we have identified EA3 as the best of the three variants, we must verify whether improvements can be made by adjusting parameters σ_1 and σ_2. Given the quality of LB^3, increasing σ_2 to values greater than 3 is counterproductive: one iteration with σ-quantile search is enough to prove that the lower bound is an upper bound. In this case, increasing σ_2 would only increase the size of the IP model constructed in the step 8 of Exact Algorithm.

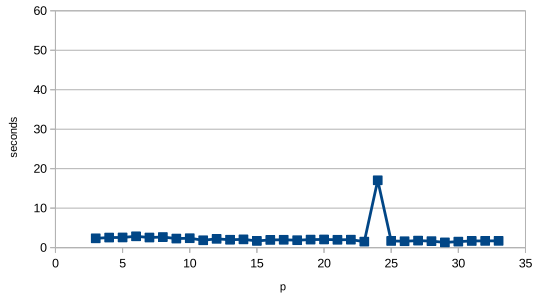
However, we can potentially improve LB^3 processing time by using other values of σ_1. Therefore, we compare the performance of algorithm EA3 on the 34 first instances from the OR-Library with different values of σ_1 in a range of values between 3 and 11. For larger instances, computational times are not reported since they exceed 5 hours for some values of σ_1. The results are given in Table 6.5, where, for each instance, the processing time of EA3 is standardised with respect to the processing time of EA3 for $\sigma_1 = 3$. Then we use the geometric mean [87] to compare the average processing time for the different values of σ_1. The experiment reveals that, with $\sigma_1 = 4$, EA3 is at least 14 percent faster than the other tested values for σ_1.

Then we perform a qualitative study to better understand the impact of p on the processing time. In Figure 6.2, the processing time of EA3 is given for three instances with 100 nodes and 200 edges, named pmed1, pmed4 and pmed5. For each instance, the curve describes the evolution of TEA3 in function of p, with p ranging from 2 to 33. For pmed2, the processing time is relatively stable for all values of p. On the contrary, the processing time to solve pmed4 and pmed5 is much more impacted by the variation of p. Precisely, processing times directly depend on the number of iterations to solve P^{rob} with the σ-quantile search (step 8). When the number of iterations is equal to 1 (which corresponds to the case $LB^3 = v(P^{rob})$) the computation time is stable. But when $LB^3 < v(P^{rob})$, the number of iterations increases (up to 5) and the computation time significantly increases.
These results underline that for a given size of instance, there is no obvious relationship between the value of p and the complexity of solving P^{rob}.

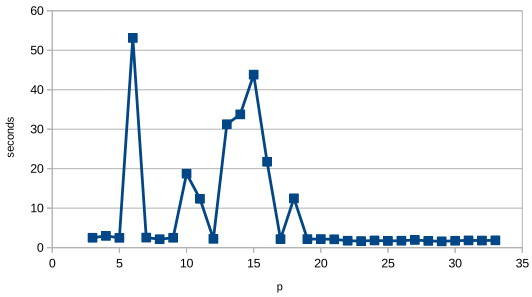
Our experimental results allow to conclude that the original algorithm EA3 is the best one. In particular we are able to precisely tune the parameters defining Algorithm 11 called EA3*. The efficiency of EA3* comes from the quality of the lower bound obtained with a very efficient 4-quantile search algorithm. With such a bound, in most cases, only one iteration of the 3-quantile search is enough to determine an optimal solution of P^{rob} in step 5.

	σ_1					
Instances	3	4	5	6	9	11
pmed1	1	0.91	1.03	1.15	1.54	1.79
pmed2	1	1.08	1.66	1.48	1.85	1.72
pmed3	1	0.98	1.32	1.34	1.18	1.94
pmed4	1	0.9	1.28	1.33	1.62	1.63
pmed5	1	0.82	1.03	0.98	1.39	1.29
pmed6	1	0.75	0.98	0.87	1.03	1.1
pmed7	1	1.46	1.84	1.91	1.68	2.92
pmed8	1	0.9	0.92	1.09	1.05	1.67
pmed9	1	0.82	0.88	1.12	1.66	1.41
pmed10	1	0.79	0.83	1.04	1.09	1.51
pmed11	1	0.81	0.9	0.89	1	1.26
pmed12	1	0.71	0.74	0.97	0.98	1.19
pmed13	1	0.82	1.47	1.47	0.93	1.69
pmed14	1	0.83	0.92	1.56	4.85	3.06
pmed15	1	1.04	1.07	1.39	1.29	1.89
pmed16	1	0.82	0.9	0.91	1.02	1.22
pmed17	1	0.91	0.98	0.99	1.08	1.27
pmed18	1	0.57	1.1	1.6	1.24	1.96
pmed19	1	0.67	0.94	0.88	1.24	3.86
pmed20	1	1.09	0.93	1.14	1.27	1.38
pmed21	1	0.86	0.9	0.94	1.45	1.24
pmed22	1	0.49	0.8	0.82	0.89	1.75
pmed23	1	0.72	1.39	2.73	2.59	1.69
pmed24	1	0.84	0.73	0.93	1.25	1.84
pmed25	1	2.26	1.64	2.57	3.68	3.73
pmed26	1	0.99	0.92	1.43	1.6	1.39
pmed27	1	0.29	1.22	0.82	0.59	0.58
pmed28	1	0.67	0.87	0.69	5.44	10.55
pmed29	1	0.6	0.53	0.61	2.25	1.33
pmed30	1	0.69	0.68	0.75	1.01	1.96
pmed31	1	1.34	1.2	1.35	1.47	1.96
pmed32	1	1.63	1.59	3.21	1.83	2.05
pmed33	1	1.33	2.21	1.23	3.77	1.38
pmed34	1	0.69	0.66	2.83	2.43	1.42
GEOMEAN	1	0.86	1.04	1.21	1.52	1.74

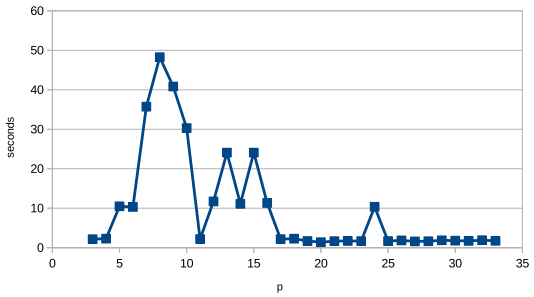
Table 6.5: Comparison of the standardized execution time of EA3 for different values of σ_1



Instance pmed2



Instance pmed4



Instance pmed5

Figure 6.2: Processing time of EA3 on three different instances for p ranging from 2 to 32

Algorithm 11 EA3*

Require: $G = (V, E)$
Ensure: Returns the optimal value and an optimal solution to P^{rob}
1: Generate $D^{rob} = \{D_1^{rob}, \ldots, D_{T^{rob}}^{rob}\}$
2: $LB^3, sol \leftarrow \sigma-$quantile search$(RP^{rob}(D^{rob}), D_1^{rob}, D_{T^{rob}}^{rob}, 4)$
3: $UB^3 \leftarrow$ value of sol for P^{rob}
4: $optValue, optSolution \leftarrow \sigma-$quantile search$(P^{rob}(D^{rob}), LB^3, UB^3, 3)$
5: Return $optValue$

To measure the advantage of using RpCP over a classical deterministic p-Center problem, we compare the value on the objective function of RpCP of an optimal solution of p-Center problem (referred to as deterministic solution) to the optimal solution value of RpCP. Let us recall that UB^1 exactly corresponds to the value, on RpCP, of a deterministic solution. Thus, we compute in Table 6.6 the ratio $UB^1/v(P^{rob})$ (none of the instances correspond to n=700 and $p = n/3$ in OR-Library). It appears that the gap can be quite significant, up to 25%. Consequently, the value of a deterministic solution can be far away from the optimal solution value of RpCP and better solutions can be found solving RpCP.

	p			
n	5	10	n/10	n/3
100	1.13	1.18	1.18	1.25
200	1.16	1.04	1.07	1.18
300	1.03	1.03	1.05	1.12
400	1.03	1.02	1.04	1.10
500	1.01	1.03	1.03	1.10
600	1.02	1.03	1.04	1.06
700	1.06	1.01	1.05	-

Table 6.6: Ratio between UB^1 and the optimal value of RpCP on OR-Library instances

In the following, we report our experimental results on randomly generated subgrid instances.

6.4.3.2 Experimental results on random subgrid instances

We have outlined the relevance of subgrids for real case applications. Thus, we chose to test our algorithms on random subgrid instances. We generated a set of 18 subgrids along the following steps. From an original undirected unit grid

of size $(l \times w)$, we generate three subgrids by randomly removing a node with probability 0.05, 0.1 and 0.2. For each subgrid, we remove isolated and leaf nodes being mandatory shelter locations. We apply this process to three grids of size $10 \times 10, 20 \times 10$ and 20×20, into nine unit subgrids $SG1, \ldots, SG9$. We then generate nine weighted subgrids $wSG1, \ldots, wSG9$ by randomly assigning length edge values, from 1 to 10, applied to the edges of $SG1, \ldots, SG9$ respectively. The data set is available at [88].

In the following, we present our synthetic analysis of the results. First, we solve P^{rob} using Exact Algorithm with $\sigma_1 = \sigma_2 = 3$ and $x = 10$. Table 6.7 gives the values of the upper and lower bounds computed for the instances $SG1, \ldots, SG9$ ordered by the number of their nodes and the value of p. Likewise, the values of the bounds that are equal to optimal values are marked with an $*$. For bounds corresponding to non-feasible solutions for `RpCP` the value is 10000. In Table 6.7, NA denotes that the values cannot be computed within 5 hours. Please note that the best values achieved for the unsolved instances can be found in Table 6.13. LB^1 is still the worst lower bound while LB^2 performs slightly better than LB^3 and both of them quite often reach optimal values. Concerning the upper bounds, we note that UB^1 and UB^3 are not so good as those of obtained for OR-Library instances.

Table 6.8 gives the value of the upper and lower bounds computed for the instances $wSG1, \ldots, wSG9$ ordered by the number of their nodes and the value of p.

In the weighted case, LB^2 and LB^3 outperform LB^1. Both lower bounds are not as good when the value of p is very low. UB^3 is the best upper bound sometimes reaching optimal value. In this case, the solution of RP^{rob} obtained with EA3 is also the optimal solution for P^{rob}.

In Tables 6.11 and 6.12, we record the processing time of our algorithms on the unit subgrids and the weighted subgrids, for the fixed values of p (5,10 and 20), and relatives values of p ($n/5$ and $n/3$). For processing time exceeding 5 hours, we record NA. For each instance, the underlined value is the best. Overall, for both unit and weighted subgrids, EA3 is the most efficient algorithm.

In Tables 6.9 and 6.10, we record the ratio $UB^1/v(P^{rob})$ on the unit subgrids and the weighted subgrids. When the deterministic solution is not feasible, the corresponding ratio value is marked ∞. We observe that the gap, up to 167%, is larger than the one recorded for OR-Library instances. In Table 6.9, when $p = n/3$, the ratio $UB^1/v(P^{rob})$ equals 1 in all instances. In this case, we also have $LB^1 = 1$ which means that the optimal solution of the p-Center problem is a dominating set. This is result in Subsection 3.2.1 : if G is a planar triangle-free graph with

Instance										
Graph	p	n	|E|	OPT	LB^1	LB^2	LB^3	UB^1	UB^2	UB^3
SG1	5	73	102	7	4	7*	6	10000	9	10000
SG1	10	73	102	4	3	4*	4*	10000	7	6
SG1	20	73	102	2	1	2*	2*	2*	4	2*
SG1	14	73	102	3	2	3*	3*	8	6	7
SG1	24	73	102	2	1	2*	2*	2*	4	2*
SG2	5	92	153	7	4	6	5	8	8	9
SG2	10	92	153	4	3	4*	4*	6	6	6
SG2	20	92	153	3	2	3*	3*	4	5	4
SG2	18	92	153	3	2	3*	3*	4	5	5
SG2	30	92	153	2	1	2*	2*	2*	4	2*
SG3	5	96	167	7	4	5	5	8	8	9
SG3	10	96	167	4	3	4*	4*	6	5	6
SG3	20	96	167	3	2	3*	3*	4	5	4
SG3	19	96	167	3	2	3*	3*	4	4	4
SG3	32	96	167	2	1	2*	2*	2*	3	2*
SG4	5	143	218	11	6	10	9	20	13	13
SG4	10	143	218	7	4	6	6	10	10	10
SG4	20	143	218	4	2	4*	4*	6	7	7
SG4	28	143	218	3	2	3*	3*	5	6	5
SG4	47	143	218	2	1	2*	2*	2*	4	2*
SG5	5	179	304	9	6	8	8	14	11	12
SG5	10	179	304	6	4	6*	5	8	8	9
SG5	20	179	304	4	2	4*	4*	7	7	6
SG5	35	179	304	3	2	3*	3*	4	5	4
SG5	59	179	304	2	1	2*	2*	2*	4	2*
SG6	5	190	332	9	6	8	8	11	12	11
SG6	10	190	332	6	4	5	5	8	9	7
SG6	20	190	332	4	3	4*	4*	6	7	6
SG6	38	190	332	3	2	3*	3*	5	5	6
SG6	63	190	332	2	1	2*	2*	2*	4	2*
SG7	5	306	483	15	9	13	12	10000	10000	10000
SG7	10	306	483	9	6	9*	8	10000	17	10000
SG7	20	306	483	6	4	6*	6*	9	11	8
SG7	61	306	483	3	2	3*	3*	10000	7	10000
SG7	102	306	483	2	1	2*	2*	2*	5	2*
SG8	5	359	620	NA	8	11	10	16	16	16
SG8	10	359	620	NA	6	7	7	11	12	12
SG8	20	359	620	NA	4	5	5	11	9	9
SG8	71	359	620	3	2	3*	3*	5	5	5
SG8	119	359	620	2	1	2*	2*	2*	4	2*
SG9	5	383	704	NA	9	11	11	16	16	18
SG9	10	383	704	NA	5	7	7	11	12	13
SG9	20	383	704	NA	4	5	3	8	8	9
SG9	76	383	704	3	2	3*	3*	4	5	4
SG9	127	383	704	2	1	2*	2*	2*	4	2*

Table 6.7: Optimal solution values and bound values for unit subgrids

Instance										
Graph	p	n	|E|	OPT	LB^1	LB^2	LB^3	UB^1	UB^2	UB^3
wSG1	5	73	102	43	20	41	36	10000	50	10000
wSG1	10	73	102	25	12	24	23	29	39	26
wSG1	20	73	102	16	8	15	16*	19	30	16*
wSG1	14	73	102	20	10	19	20*	29	31	22
wSG1	24	73	102	13	8	13*	13*	20	25	13*
wSG2	5	92	153	29	17	29*	27	35	36	43
wSG2	10	92	153	20	11	20*	20*	25	27	26
wSG2	20	92	153	15	7	15*	15*	18	23	15*
wSG2	18	92	153	16	7	16*	16*	19	23	19
wSG2	30	92	153	12	5	12*	12*	15	20	12*
wSG3	5	96	167	27	14	26	25	35	35	39
wSG3	10	96	167	19	10	19*	19*	25	29	22
wSG3	20	96	167	15	7	15*	15*	20	20	15*
wSG3	19	96	167	15	7	15*	15*	21	22	15*
wSG3	32	96	167	11	5	11*	11*	15	19	11*
wSG4	5	143	218	43	27	42	42	56	53	81
wSG4	10	143	218	31	16	30	29	39	41	41
wSG4	20	143	218	22	11	21	21	29	35	29
wSG4	28	143	218	17	8	17*	17*	25	29	25
wSG4	47	143	218	13	6	13*	13*	25	24	13*
wSG5	5	179	304	39	25	38	37	51	48	45
wSG5	10	179	304	29	17	29*	28	33	39	33
wSG5	20	179	304	22	11	21	21	28	30	23
wSG5	35	179	304	17	8	17*	17*	24	27	29
wSG5	59	179	304	13	6	12	13*	24	24	13*
wSG6	5	190	332	40	26	40*	39	48	55	45
wSG6	10	190	332	30	17	29	28	44	45	49
wSG6	20	190	332	21	11	21*	21*	33	37	40
wSG6	38	190	332	17	8	16	17*	21	27	17*
wSG6	63	190	332	13	6	13*	13*	19	22	13*
wSG7	5	306	483	67	38	65	59	10000	10000	10000
wSG7	10	306	483	44	26	44*	43	10000	88	10000
wSG7	20	306	483	32	17	32*	31	10000	62	10000
wSG7	61	306	483	17	8	17*	17*	10000	35	10000
wSG7	102	306	483	13	6	13*	13*	10000	30	10000
wSG8	5	359	620	53	34	47	44	69	70	71
wSG8	10	359	620	38	23	35	35	51	53	52
wSG8	20	359	620	28	16	27	27	35	40	32
wSG8	71	359	620	17	8	17*	17*	22	29	19
wSG8	119	359	620	13	6	13*	13*	17	24	13*
wSG9	5	383	704	58	34	47	46	79	68	81
wSG9	10	383	704	38	23	36	36	51	51	59
wSG9	20	383	704	28	17	27	27	39	44	35
wSG9	76	383	704	17	8	16	17*	22	29	19
wSG9	127	383	704	13	6	13*	13*	17	24	13*

Table 6.8: Optimal solution values and bound values for weighted subgrids

	p			
Instance	5	10	n/5	n/3
SG1	∞	∞	2.67	1
SG2	1.14	1.5	1.33	1
SG3	1.14	1.5	1.33	1
SG4	1.82	1.43	1.67	1
SG5	1.56	1.33	1.33	1
SG6	1.22	1.33	1.67	1
SG7	∞	∞	∞	1
SG8	NA	NA	∞	1
SG9	NA	NA	1.33	1

Table 6.9: Ratio between UB^1 and optimal solution value of `RpCP` on unit subgrids

	p			
Instance	5	10	n/5	n/3
wSG1	∞	1.16	1.45	1.54
wSG2	1.21	1.25	1.19	1.25
wSG3	1.3	1.32	1.4	1.36
wSG4	1.3	1.26	1.47	1.92
wSG5	1.31	1.14	1.41	1.85
wSG6	1.2	1.47	1.24	1.46
wSG7	∞	∞	∞	∞
wSG8	1.3	1.34	1.29	1.31
wSG9	1.36	1.34	1.29	1.31

Table 6.10: Ratio between UB^1 and optimal solution value of `RpCP` on weighted subgrids

no leaf nodes and C a dominating set, then $r(C) \leq 2$. Moreover, when $p = n/3$, there is no solution with radius equals to 1. So, the deterministic solution is an optimal solution for `RpCP`.

Instance						
Graph	p	n	|E|	T EA1	T EA2	T EA3
SG1	5	73	102	8.62	6.42	4.62
SG1	10	73	102	4.48	6.14	2.6
SG1	14	73	102	3.37	6.25	2.43
SG1	20	73	102	0.84	5.42	0.31
SG1	24	73	102	0.81	5.58	0.32
SG2	5	92	153	15.02	19.87	13.47
SG2	10	92	153	5.98	11.83	5.46
SG2	18	92	153	2.71	10.19	3.8
SG2	20	92	153	2.86	10.01	2.3
SG2	30	92	153	1.43	10.23	0.59
SG3	5	96	167	29.38	38.74	31
SG3	10	96	167	7.54	9.74	6.08
SG3	19	96	167	3.21	9.82	2.66
SG3	20	96	167	3.28	11.87	2.61
SG3	32	96	167	1.66	10.56	0.67
SG4	5	143	218	61.04	72.64	32.84
SG4	10	143	218	29.54	52.4	27.98
SG4	20	143	218	14.57	39.34	13.16
SG4	28	143	218	13.34	36.56	10.61
SG4	47	143	218	4.75	37.96	1.42
SG5	5	179	304	188.98	281.31	317.06
SG5	10	179	304	70.04	80	57.87
SG5	20	179	304	37.37	72.1	23.35
SG5	35	179	304	15.16	69.52	12.48
SG5	59	179	304	8.8	73.25	2.61
SG6	5	190	332	399.16	855.92	486.3
SG6	10	190	332	143.33	236.57	101.38
SG6	20	190	332	44.3	88.26	30.09
SG6	38	190	332	31.71	83.41	25.69
SG6	63	190	332	10.74	87.64	3.13
SG7	5	306	483	1298.46	1364.17	1683.74
SG7	10	306	483	585.89	394.51	924.87
SG7	20	306	483	205.19	354.8	110.61
SG7	61	306	483	309.12	304.88	136.95
SG7	102	306	483	37.03	324.04	8.53
SG8	5	359	620	NA	NA	NA
SG8	10	359	620	NA	NA	NA
SG8	20	359	620	NA	NA	NA
SG8	71	359	620	355.2	675	258.2
SG8	119	359	620	88.4	791.4	40.3
SG9	5	383	704	NA	NA	NA
SG9	10	383	704	NA	NA	NA
SG9	20	383	704	NA	NA	NA
SG9	76	383	704	196.3	776.4	124.4
SG9	127	383	704	95	837	21.9

Table 6.11: Processing time of each variant of the Exact Algorithm on the unit subgrids

Instance						
Graph	p	n	|E|	T EA1	T EA2	T EA3
wSG1	5	73	102	16.19	15.51	18.27
wSG1	10	73	102	6.06	9.36	3.24
wSG1	14	73	102	4.6	8.57	1.92
wSG1	20	73	102	3.17	8.01	0.54
wSG1	24	73	102	3.1	6.61	0.43
wSG2	5	92	153	22.48	24.17	24.09
wSG2	10	92	153	12.98	18.28	7.99
wSG2	18	92	153	8.41	12.26	4.35
wSG2	20	92	153	7.22	11.92	0.86
wSG2	30	92	153	5.96	12.17	0.66
wSG3	5	96	167	28.39	33.72	27.66
wSG3	10	96	167	9.67	16.74	5.36
wSG3	19	96	167	7.46	13.47	0.9
wSG3	20	96	167	6.77	13.28	0.9
wSG3	32	96	167	5.38	13.73	0.92
wSG4	5	143	218	35.29	74.06	40.19
wSG4	10	143	218	53.41	64.43	45.99
wSG4	20	143	218	33.56	56.6	25.62
wSG4	28	143	218	25.78	45.52	11.73
wSG4	47	143	218	23.7	45	2.29
wSG5	5	179	304	100.58	219.64	134.92
wSG5	10	179	304	166.78	89.13	75.09
wSG5	20	179	304	98.51	150.31	94.4
wSG5	35	179	304	40.31	80.68	22.47
wSG5	59	179	304	50.68	100.62	3.81
wSG6	5	190	332	188.65	137.97	97.1
wSG6	10	190	332	288.23	606.18	393.31
wSG6	20	190	332	108.05	161.24	70.87
wSG6	38	190	332	79.25	155.49	4.13
wSG6	63	190	332	47.53	99.98	4.24
wSG7	5	306	483	1168.59	1902.13	2553.12
wSG7	10	306	483	845.52	701.73	368.59
wSG7	20	306	483	769.56	728.43	886.37
wSG7	61	306	483	648.31	394.3	250.37
wSG7	102	306	483	602.1	396.6	175.92
wSG8	5	359	620	NA	NA	17843
wSG8	10	359	620	NA	10098.49	9800.68
wSG8	20	359	620	8700.72	8313.83	17374.92
wSG8	71	359	620	518.06	772.39	201.35
wSG8	119	359	620	356.19	858.28	41.93
wSG9	5	383	704	13189.6	NA	NA
wSG9	10	383	704	NA	NA	15588.2
wSG9	20	383	704	6358.69	15998.37	9624.58
wSG9	76	383	704	1226.05	2258.24	320.34
wSG9	127	383	704	561.36	1009.85	29.16

Table 6.12: Processing time of each variant of the Exact Algorithm on the weighted subgrids

Instance					
Graph	p	n	|E|	Best Value = UB	LB
SG8	5	359	620	14	11
SG8	10	359	620	9	7
SG8	20	359	620	7	5
SG9	5	383	704	13	11
SG9	10	383	704	11	7
SG9	20	383	704	7	5

Table 6.13: Best results obtained on the unsolved instances of unit subgrids

Conclusion

In this chapter we introduce mathematical programming methods for `RpCP` and `PpCP`. Based on MIP formulations for `Min p-Center`, we propose three MIP formulations each for `RpCP` and `PpCP`. We also propose an IP formulation for `k-RCP`, based on a set covering approach. The size of the instances generated by our formulations are however too large in practice. To propose a tractable solution method for `RpCP`, we develop exact algorithms based on P^{rob} using a decomposition scheme. We present three different methods to compute lower and upper bounds for `RpCP`. To lessen the size of an instance and the number of iterations, we propose a generalization of the binary search: the σ-quantile search. We test different algorithms, each of which integrates a different method to generate lower and upper bound. Our experimental results shows that algorithm EA3* is the best one: its efficiency comes from the quality of the lower bound obtained with a very efficient 4-quantile search algorithm for RP^{rob} and the low computation time. We have tested our algorithm and generated the different bounds on the OR library and a domestic set of uniform and edge-weighted subgrids. Overall, the RP^{rob}-based algorithm remains the most efficient in terms of computing time. In addition, our experiments highlight the performance of a `RpCP` optimal solution compared to a `Min p-Center` optimal solution: we compared the value of an optimal solution for `Min p-Center` taken for `RpCP` with the optimal value. Even if we exclude unfeasible cases, it appears that the gap can be quite significant, up to 92% for instances with more than 100 vertices.

CHAPTER 7

CONCLUSION

In this concluding chapter, we first summarize the contribution of this thesis. Next, we discuss possible applications and variants of our work and some open questions for future research.

7.1 Context and contributions

In the prevention phase of wildfire, an important problem is to determine shelters location in a given territory in order to minimize the evacuation time given different fire scenarios. In this thesis, we propose a new two-stage model to locate such shelters on territories with low density of people on it. We present in Chapter 2 the model as well as problems `RpCP` and `PpCP`. The uncertainty of fire outbreaks is introduced taking into account a finite set of fire scenarios. A scenario defines a fire outbreak on a single zone with the main consequence of modifying the operational paths. At most p shelters are to be located on vertices in the preparedness phase, thus our solution is a p-set. In the response phase, that is when a scenario occurs, everyone in the territory must reach a shelter following an evacuation path. The main novelty of our model is the UP evacuation strategy. With this evacuation strategy, we propose a robust representation of the evacuation of the source vertex: as there may be different people in different places within the source vertex, and given the pressure induced by the imminent danger and the disruptions caused by the fire on the internal transport network in the source vertex, we consider that any of the paths leaving the vertex may be used for evacuation. We introduce then two new facility location problems, incorporating the UP evacuation strategy, as non deterministic variants of the `Min p-Center` problem. The `PpCP` problem for a risk environment, and the `RpCP` problem for an uncertainty environment. The

set of feasible solutions corresponds to solutions ensuring, for all scenarios and all vertices, finite evacuation distance. A direct result is that a shelter must be located on each pending vertex.

To our knowledge, it is the first attempt to model the evacuation process in such manner for a shelter location problem. This model differentiates itself from most existing models in that we apply indeterminacy on the graph structure and not on the values of the edges between the vertices. By doing so, we consider that we can better apprehend the impact of the fire on the evacuations distances.

In Chapter 3, we study the hardness of RpCP and PpCP. Regarding RpCP, we consider the version of RpCP with a constant evacuation radius threshold, $\texttt{RpCP}_k$. We give various reduction from Min Vertex Cover and Min Dominating Set to $\texttt{RpCP}_k$ for different values of k. It follows that:

- on all hereditary classes of graphs for which the decision version of Min Vertex Cover problem is NP-complete: $\texttt{RpCP}_1$ is NP-complete and RpCP is NP-hard.
- on all hereditary classes of graphs for which the decision version of Min Vertex Cover problem is polynomial-time solvable: $\texttt{RpCP}_1$ is polynomial-time solvable.
- in all triangle-free classes of uniform graphs and of minimum degree 2, for which the decision version of Min Dominating Set problem is NP-complete: $\texttt{RpCP}_2$ is NP-complete and RpCP is NP-hard.
- in all triangle-free classes of uniform graphs and of minimum degree 2, for which the decision version of Min Dominating Set problem is polynomial-time solvable: $\texttt{RpCP}_2$ is polynomial-time solvable.
- in subgrids with vertices of degree 2 or 3, $\texttt{RpCP}_2$ is NP-complete and RpCP is NP-hard.

For the robust case, our strongest result, the NP-hardness of RpCP in subgrids with vertices of degree 2 or 3, is obtained by a reduction from Min Vertex Cover to $\texttt{RpCP}_2$, through the relation between $\texttt{RpCP}_2$ and Min Dominating Set. In addition, we outlined the existence of graph classes on which $\texttt{RpCP}_1$ is hard but $\texttt{RpCP}_2$ is trivial. This seems to confirm that a reduction from $\texttt{RpCP}_k$ to $\texttt{RpCP}_k + 1$ does not exist in the general case and consequently, the hardness of $\texttt{RpCP}_k$ on a given graph class requires to be studied for any value of k and cannot be deduced, in general, from hardness results dealing with different values of k.

Regarding PpCP, we give two non-approximation results. First, we prove there is no polynomial time approximation for PpCP guaranteeing a ratio less than $\frac{20}{19}$ for

bipartite planar graphs of degrees 2 or 3. In particular, in Section 3, we revisit the previous reduction to prove that PpCP is not approximable with a ratio less than $\frac{56}{55}$ on subgrids of degree at most 3.

In Chapter 4, we study polynomial and approximation results for PpCP. We give an explicit solution for the uniform case on paths. In this case, a p-set is characterized by the list of lengths of segments between two consecutive centers. A solution is then called balanced if the maximum difference between two segment lengths is minimized and it is monotone if the sequence of segment lengths is monotone. We then show that a monotone balanced solution is optimal for PpCP. Even though the result is not surprising, the proof was surprisingly non-trivial. The main idea is to express the objective function as the sum of two parts (contribution of the centers and the non-center vertices) and simultaneously bound the objective function by another, objective function more easily computable. We prove then independently that a balanced monotone solution minimizes simultaneously both terms. This results is however not valid on edge-weighted path, as we show it by an counterexample.

In addition, we propose some approximation results for PpCP. We propose a $4\overline{deg}(G) + 2$-approximation for graphs of bounded degree and with edge lengths in $[\ell, 2\ell]$. It results in a 3-approximation on edge-weighted trees. This approximation result holds for a class of instances on which all our hardness results apply: it provides a gap between constant approximation ratios and the hardness in approximation results we have obtained. In the process, we have introduced the Min MAC p-Center problem, which is the problem of finding a p-set of minimum radius among the set of feasible solutions for PpCP. We have shown that this problem is 2-approximable and that this is the best possible constant approximation ratio. It is also polynomial on trees.

In Chapter 5, we propose polynomial algorithms to solve RpCP on acyclic edge-weighted graphs. Our algorithms are based on the auxiliary problem k-RCP. By performing a binary search on the candidate values of k, we can find an optimal solution for RpCP in $O(f(n)\log(n))$, where $f(n)$ is the complexity of an algorithm for k-RCP. By generalizing the approach used for PpCP on paths, we propose iterative algorithms for k-RCP that locate shelters from bottom up. This allows us to develop refined polynomial algorithms for k-RCP on paths, extended stars, caterpillars and trees.

In Chapter 6, we propose mathematical methods to solve RpCP and PpCP. We propose different MIP formulations for RpCP and PpCP and an IP formulation for k-RCP. These programs appear however to be quickly intractable with the size of

the graph. We develop therefore an efficient algorithm for RpCP based on formulation P^{rob}. It relies on the generation of lower and upper bounds, coupled with a binary search. Experimental results on general graphs and subgrids outline the efficiency of our approach.

We summarize in Table 7.1 most of our results.

	RpCP	PpCP
Uniform path	Exact solution*	Exact solution
Edge-weighted path, Star, Caterpillar	$O(n^2)$	Unknown complexity
Tree	$O(n^2)$ even in the edge-weighted case	3-approximable in the uniform case NP-hardness unknown
Graph with bounded edges and bounded degrees	$(4\overline{deg}(G)+2)$-approximable*	$(4\overline{deg}(G)+2)$-approximable
Bipartite-planar graphs of degrees 2 or 3	NP-hard	Non approximable under $\frac{20}{19}$
Subgrids of degrees 2 or 3	NP-hard	Non approximable under $\frac{56}{55}$

Table 7.1: Hardness, approximation and non-approximation results.

* These RpCP results can be straightforwardly induced from the corresponding PpCP proofs.

7.2 Open problems

Our work still has a high potential for future improvements, both from a theoretical and a practical perspective. We conclude our discussion by enumerating some explicit future research directions.

7.2.1 Theoretical open questions

From a theoretical perspective, a first open question concerns the complexity of solving PpCP on edge-weighted acyclic graphs. While a 3-approximation result on trees with bounded edge is ensured, we have few information on the structure of an

optimal solution for PpCP on acyclic graphs, not even on an edge-weighted path. A research track would be to analyze the quality of the optimal RpCP solution, polynomially computable on these graphs with respect to PpCP. We know that an optimal RpCP solution is not always optimal for PpCP, however we have poor results yet on the structural relations between the optimal solutions of both problems. The quality of an optimal solution for RpCP used for PpCP, and vice versa, should be analyzed for different classes of graphs.

Another research track to identify solutions for PpCP, would be to restrict our problem to the local probabilistic radius, introduced in Section 4.1. The local probabilistic radius, based on the local evacuation radius, enables us to focus our analysis on the maximum evacuation distance induced by the source vertex and its neighborhood. The relevance of this approach has also been displayed in Section 6.4. Indeed, RP^{rob} corresponds to the problem RpCP restricted to the evacuation distance of the source vertex solely. While RP^{rob} is a relaxed variant of the local robust radius (the robust version of the local probabilistic radius), it often provides a tight lower bound for RpCP. The local evacuation radius has also a practical interest, which we will come back on later. Finally, as the approximation gap is still large, there is potential for further approximation and/or non-approximation results.

For RpCP, we started implementing our polynomial algorithms for acyclic graphs. It would be interesting to assess their efficiency with experimental results. In addition, we did not have time to design specific approximation algorithm for RpCP on planar graphs and subgrids. A good starting point would be to consider graphs with exactly one pending vertex, and no other minimal articulation component. In this setup, the problem would be to decide the location of the other shelters. Note that the NP-hardness of RpCP is not established on such graphs. This approach is inspired by the acyclic graphs approximation algorithms. Indeed, as shelters must be located on the pending vertices, these pending vertices are the foundation of our solutions.

An additional meaningful research track would be to improve our exact IP-based algorithm for RpCP and to develop an efficient exact algorithm for PpCP. On both problems, we should be able to implement a Benders decomposition to tackle the issue raised by the size of the problems. Given the quality observed of an optimal solution for RP^{rob} in Chapter 6, RP^{rob} can be taken as the initial master problem. In addition, it would be interesting to determine and generate constraints specific to the class of the graph considered.

An interesting line of research would be to consider distances verifying triangles inequalities, and analyze if they induce structural properties for our problems.

Finally, some initial hypothesis (shelter property, scenario definition, etc.) can be adapted to generate different variants of our problems. The relevance of these

variants mainly depends on their practical application. We list them next.

7.2.2 Practical open questions

From a practical perspective, our model can be adapted to different fire configurations. We list some of them. Note that our mathematical programming model could easily integrate these variants, but our combinatorial results may change significantly.

1. We can take into account different constraints or consequences of locating a shelter on an area.

   - The territory may limit the location of shelters to some specific areas. In this case, we can define initially the subset of vertices on which it is possible to locate a shelter. This variant includes our model as specific case, thus our hardness results remain valid. However, further research is required to extend our polynomial and approximation results to this variant.
   - Building a shelter in an area may involve securing completely the corresponding area, in which case a fire can not occur in an area with a shelter located on it. We use an auxiliary problem that is relatively close to this problem in Section 4.1, when we use the objective function $E_{\bar{C}}$. $E_{\bar{C}}$ is the sum of the evacuation radius for all scenarios $s \in \bar{C}$, weighted by the total number of scenarios (and not by $|\bar{C}|$). Further research is required to reduce this variant to our model and extend our results to it.
   - On the contrary, the shelter located on an area may not be sufficient to secure the area if the fire starts within the area. The consequence on the evacuation distance computation formula is straightforward. Further research is required to reduce this variant to our model and extend our results to it.

2. We can take into account the fire spread, in a static way, by considering scenarios associated to a subset of vertices in fire. In this case, all vertices in fire are no longer reachable.

3. Under some circumstances, only a partial evacuation is needed. When a fire occurs on an area, only the corresponding area must be evacuated. A close variant is the case where solely the source area and its neighborhood must be evacuated. The latter corresponds to the approach we initiated Section 4.1 with the local evacuation radius. This strengthens our interest

for this variant already discussed in the previous section. Further research is required to reduce this variant to our model and extend our results to it.

7.3 Production

The present thesis is divided into five main chapters. At the time of writing two articles have been published, one has been submitted, and another is under preparation.

7.3.1 List of publications

7.3.1.1 Articles published in EURO journals

- **Haddad, Marcel A.** and Demange, Marc and Gabrel, Virginie and Murat, Cécile. 'A robust p-Center problem under pressure to locate shelters in wildfire context." **EURO Journal on Computational Optimization** 8.2 (2020): 103-139.

7.3.1.2 Articles submitted

- **Haddad, Marcel A.** and Demange, Marc and Murat, Cécile. 'Hardness and approximation of the Probabilistic p-Center problem under Pressure" (submitted September 2020)

7.3.1.3 Proceedings published

- **Haddad, Marcel A.** and Demange, Marc and Murat, Cécile., 'The probabilistic k-center problem'. Proceedings of the GEOSAFE Workshop on Robust Solutions for Fire Fighting, RSFF 2018, L'Aquila, Italy, July 19-20, 2018.

7.3.2 Conferences

Conference Wildfire Conference | Addressing the Challenges of Bushfire Management,
11-15 November 2019, Melbourne, Australia

Title Variants of the Robust p-Center under Pressure problem, A new approach for robust shelter location

Authors Haddad, Marcel A. and Demange, Marc and Gabrel, Virginie and Murat, Cécile.

Conference 6th International Fire Behavior and Fuels Conference, April 29 – May 3 2019 Marseille, France

Title New mathematical models to locate shelters in wild fire context.

Authors Haddad, Marcel A. and Demange, Marc and Gabrel, Virginie and Murat, Cécile.

Conference ROADEF 2019
February 19 - 21 2019, Le Havre, France

Title Formulations PLNE pour le problème du p-Centre non déterministe

Authors Haddad, Marcel A. and Demange, Marc and Murat, Cécile.

Conference RSFF 2018, International workshop on robust solutions for fire fighting, 19 - 20 July 2018. L'Aquila, Italy

Title The probabilistic k-center problem

Authors Haddad, Marcel A. and Demange, Marc and Murat, Cécile.

Conference 5th International Combinatorics Conference
December 4–9 2017, Monash University, Melbourne, Australia

Title The k-center problem in graphs with uncertainty on vertices

Authors Haddad, Marcel A. and Demange, Marc and Murat, Cécile.

APPENDIX A

LIST OF PROBLEMS

`Min p-Center`

Instance:	An edge-weighted graph $G = (V, E)$, an integer p
Feasible solutions:	A p-set i.e., a set $C \subseteq V$ of size at most p
Objective:	Find a p-set C of minimum radius $r(C)$.

`RpCP`

Instance:	An edge-weighted graph $G = (V, E)$, a set of scenarios S, an integer p
Feasible solutions:	A p-set $C \subseteq V$ with $\mathbf{R}(C)$ finite
Objective:	Find a solution of minimum robust radius if it exists.

`PpCP`

Instance:	An edge-weighted graph $G = (V, E)$, a set of scenarios S and a vector Π of probabilities associated to the scenarios of S, an integer p
Feasible solutions:	A p-set $C \subseteq V$ with $\mathbf{E}(C)$ finite
Objective:	Find a solution of minimum probabilistic radius if it exists.

`k-RCP`

Instance:	An edge-weighted graph $G = (V, E)$, a set of scenarios S, an integer k
Feasible solutions:	A set $C \subseteq V$ of robust radius $\mathbf{R}(C) \leq k$.
Objective:	Find a feasible solution of minimum size.

`Min Vertex Cover`

Instance:	A graph $G = (V, E)$
Feasible solutions:	A vertex cover i.e., a set $U \subseteq V$ such that every edge of E is incident to at least one vertex of U
Objective:	Find a vertex cover of minimum size ($\tau(G)$).

`Min Dominating Set`

Instance:	A graph $G = (V, E)$
Feasible solutions:	A dominating set i.e., a set $D \subseteq V$ such that every vertex of $V \setminus D$ is adjacent to a vertex of D.
Objective:	Find a dominating set of minimum size ($\gamma(G)$).

`Min Set Cover`

Instance:	A set system $(\Omega, \mathcal{Z})$ with $\cup_{Z \subset \mathcal{Z}} Z = \Omega$
Feasible solutions:	A set cover i.e., a set $\mathcal{Y} \in \mathcal{Z}$ such that $\cup_{Y \in \mathcal{Y}} Y = \Omega$
Objective:	Find a set cover of minimum size.

`Min r-Dominating Set`

Instance:	A graph $G = (V, E)$, an integer r
Feasible solutions:	A set $C \subset V$ such that $r(C) \leq r$
Objective:	Find a feasible solution of minimum size.

`Min MAC p-Center`

Instance:	A graph $G = (V, E)$, an integer p
Feasible solutions:	$\mathcal{C}_p(G)$ whose elements are called MAC p-set
Objective:	Find a MAC p-set of minimum radius $r(C)$.

`Min Partial p-Center`

Instance:	An edge-weighted graph $G = (V, E)$, a set $U \subset V$, an integer p
Feasible solutions:	Any p-set
Objective:	Find a p-set C of minimum partial radius.

BIBLIOGRAPHY

[1] Intergovernmental Panel On Climate Change. *Climate change 2013: the physical science basis: Working Group I contribution to the Fifth assessment report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2014.

[2] Douglas Paton, Petra T Buergelt, Sarah McCaffrey, and Fantina Tedim. *Wildfire hazards, risks, and disasters*. Elsevier, 2014.

[3] Jerry Williams. Exploring the onset of high-impact mega-fires through a forest land management prism. *Forest Ecology and Management*, 294:4–10, 2013.

[4] U.S. Fire Administration. `https://www.usfa.fema.gov/data/statistics/`.

[5] Country Fire Authority Australia. Center for disaster philanthropy. `https://disasterphilanthropy.org/disaster/2019-australian-wildfires/`.

[6] GEO-SAFE project. `http://geosafe.lessonsonfire.eu/`.

[7] David L Martell. A review of recent forest and wildland fire management decision support systems research. *Current Forestry Reports*, 1(2):128–137, 2015.

[8] Jane A Bullock, George D Haddow, and Damon P Coppola. *Introduction to emergency management*. Butterworth-Heinemann, 2017.

[9] Country Fire Authority. Community fire refuges. `https://www.cfa.vic.gov.au/plan-prepare/community-fire-refuges`.

[10] Thomas J Cova, Frank A Drews, Laura K Siebeneck, and Adrian Musters. Protective actions in wildfires: evacuate or shelter-in-place? *Natural Hazards Review*, 10(4):151–162, 2009.

[11] Kent Steer, Ermyas Abebe, Mahathir Almashor, Anton Beloglazov, and Xu Zhong. On the utility of shelters in wildfire evacuations. *Fire safety journal*, 94:22–32, 2017.

[12] Gilbert Laporte, Stefan Nickel, and Francisco Saldanha-da Gama. Introduction to location science. In *Location science*, pages 1–21. Springer, 2019.

[13] Charles S ReVelle and Horst A Eiselt. Location analysis: A synthesis and survey. *European journal of operational research*, 165(1):1–19, 2005.

[14] M Teresa Melo, Stefan Nickel, and Francisco Saldanha-Da-Gama. Facility location and supply chain management–a review. *European journal of operational research*, 196(2):401–412, 2009.

[15] Oded Kariv and S Louis Hakimi. An algorithmic approach to network location problems. i: The p-centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979.

[16] Jonathan Rosenhead, Martin Elton, and Shiv K Gupta. Robustness and optimality as criteria for strategic decisions. *Journal of the Operational Research Society*, 23(4):413–431, 1972.

[17] Pau Costa Foundation. `www.paucostafoundation.org/`.

[18] INFOCA. `www.juntadeandalucia.es/medioambiente/site/portalweb/`.

[19] Service d'Incendie et de Secours de la Haute Corse. `www.sdis2b.fr`.

[20] Country Fire Authority. `www.cfa.vic.gov.au`.

[21] Bernhard Korte, Jens Vygen, B Korte, and J Vygen. *Combinatorial optimization*, volume 2. Springer, 2012.

[22] Vangelis T Paschos. *Complexité et approximation polynomiale: optima locaux et rapport différentiel*. Hermès Science publications, 2004.

[23] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[24] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

[25] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[26] Peter J Slater. R-domination in graphs. *Journal of the ACM (JACM)*, 23(3):446–450, 1976.

[27] M. Demange, M.A. Haddad, and C. Murat. The probabilistic k-center problem. In Gabriele Di Stefano and Alfredo Navarra, editors, *Proceedings of the GEOSAFE Workshop on Robust Solutions for Fire Fighting, RSFF 2018, L'Aquila, Italy, July 19-20, 2018*, volume 2146 of *CEUR Workshop Proceedings*, pages 62–74. CEUR-WS.org, 2018.

[28] Marc Demange, Virginie Gabrel, Marcel A Haddad, Cécile Murat, et al. A robust p-center problem under pressure to locate shelters in wildfire context. *EURO Journal on Computational Optimization*, 8(2):103–139, 2020.

[29] ES Sheppard. A conceptual framework for dynamic location—allocation analysis. *Environment and Planning a*, 6(5):547–564, 1974.

[30] Vedat Bayram. Optimization models for large scale network evacuation planning and management: A literature review. *Surveys in Operations Research and Management Science*, 21(2):63–84, 2016.

[31] Panos Kouvelis and Gang Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 2013.

[32] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[33] Alfred Weber. *Theory of the Location of Industries*. University of Chicago Press, 1929.

[34] David Fearon. Alfred weber: theory of the location of industries, 1909. *Ŕ Center for Spatially Integrated Social Science*, 2006.

[35] S Louis Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459, 1964.

[36] Aakil M Caunhye, Xiaofeng Nie, and Shaligram Pokharel. Optimization models in emergency logistics: A literature review. *Socio-economic planning sciences*, 46(1):4–13, 2012.

[37] Gilbert Laporte, Stefan Nickel, and Francisco Saldanha da Gama. *Location science*, volume 528. Springer, 2015.

[38] Lawrence V Snyder. Facility location under uncertainty: a review. *IIE transactions*, 38(7):547–564, 2006.

[39] S Louis Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations research*, 13(3):462–475, 1965.

[40] Edward Minieka. The m-center problem. *Siam Review*, 12(1):138–139, 1970.

[41] Mikkel Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *Journal of the ACM (JACM)*, 46(3):362–394, 1999.

[42] Gabriel Y Handler. Minimax location of a facility in an undirected tree graph. *Transportation Science*, 7(3):287–293, 1973.

[43] Wen-Lian Hsu and George L Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.

[44] Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.

[45] Martin E Dyer and Alan M Frieze. A simple heuristic for the p-centre problem. *Operations Research Letters*, 3(6):285–288, 1985.

[46] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[47] Tomás Feder and Daniel Greene. Optimal algorithms for approximate clustering. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 434–444. ACM, 1988.

[48] Jesus Garcia-Diaz, Rolando Menchaca-Mendez, Ricardo Menchaca-Mendez, Saúl Pomares Hernández, Julio César Pérez-Sansalvador, and Noureddine Lakouari. Approximation algorithms for the vertex k-center problem: Survey and experimental evaluation. *IEEE Access*, 7:109228–109245, 2019.

[49] Gabriel Y Handler. Minimax network location: theory and algorithms. Technical report, Cambridge, Mass.: Massachusetts Institute of Technology, Flight ..., 1974.

[50] Mark S Daskin. *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons, 1995.

[51] Sourour Elloumi, Martine Labbé, and Yves Pochet. A new formulation and resolution method for the p-center problem. *INFORMS Journal on Computing*, 16(1):84–94, 2004.

[52] Hatice Calik and Barbaros C Tansel. Double bound method for solving the p-center location problem. *Computers & Operations Research*, 40(12):2991–2999, 2013.

[53] Nenad Mladenović, Martine Labbé, and Pierre Hansen. Solving the p-center problem with tabu search and variable neighborhood search. *Networks: An International Journal*, 42(1):48–64, 2003.

[54] Wayne Pullan. A memetic genetic algorithm for the vertex p-center problem. *Evolutionary computation*, 16(3):417–436, 2008.

[55] Said Salhi and A Al-Khedhairi. Integrating heuristic information into exact methods: The case of the vertex p-centre problem. *Journal of the Operational research society*, 61(11):1619–1631, 2010.

[56] Joseph S Martinich. A vertex-closing approach to the p-center problem. *Naval Research Logistics (NRL)*, 35(2):185–201, 1988.

[57] Borut Robič and Jurij Mihelič. Solving the k-center problem efficiently with a dominating set algorithm. *Journal of computing and information technology*, 13(3):225–234, 2005.

[58] Burçin Bozkaya and Barbaros Tansel. A spanning tree approach to the absolute p-center problem. *Location Science*, 6(1-4):83–107, 1998.

[59] Derya Celik Turkoglu and Mujde Erol Genevois. A comparative survey of service facility location problems. *Annals of Operations Research*, pages 1–70, 2019.

[60] Zuo-Jun Max Shen, Roger Lezhou Zhan, and Jiawei Zhang. The reliable facility location problem: Formulations, heuristics, and approximation algorithms. *INFORMS Journal on Computing*, 23(3):470–482, 2011.

[61] Eduardo Álvarez-Miranda, Elena Fernández, and Ivana Ljubić. The recoverable robust facility location problem. *Transportation Research Part B: Methodological*, 79:93–120, 2015.

[62] Shiva Chaudhuri, Naveen Garg, and R Ravi. The p-neighbor k-center problem. *Information Processing Letters*, 65(3):131–134, 1998.

[63] Doron Chen and Reuven Chen. Optimal algorithms for the α-neighbor p-center problem. *European Journal of Operational Research*, 225(1):36–43, 2013.

[64] Rongbing Huang, Seokjin Kim, and Mozart BC Menezes. Facility location for large-scale emergencies. *Annals of Operations Research*, 181(1):271–286, 2010.

[65] B. Du, H. Zhou, and R. Leus. A two-stage robust model for a reliable p-center facility location problem. *Applied Mathematical Modelling*, 77:99–114, 2020.

[66] Binay Bhattacharya, Tsunehiko Kameda, and Zhao Song. Minmax regret 1-center algorithms for path/tree/unicycle/cactus networks. *Discrete Applied Mathematics*, 195:18–30, 2015.

[67] Igor Averbakh and Oded Berman. Minimax regret p-center location on a network with demand uncertainty. *Location Science*, 5(4):247–254, 1997.

[68] M. Taghavi and H. Shavandi. The p-center problem under uncertainty. *Journal of Industrial and Systems Engineering*, 6(1):48–57, 2012.

[69] Igor Averbakh and Oded Berman. Algorithms for the robust 1-center problem on a tree. *European Journal of Operational Research*, 123(2):292–302, 2000.

[70] Luisa I Martínez-Merino, Maria Albareda-Sambola, and Antonio M Rodríguez-Chía. The probabilistic p-center problem: Planning service for potential customers. *European Journal of Operational Research*, 262(2):509–520, 2017.

[71] Igor Averbakh. Complexity of robust single facility location problems on networks with uncertain edge lengths. *Discrete Applied Mathematics*, 127(3):505–522, 2003.

[72] C. Lu. Robust weighted vertex p-center model considering uncertain data: An application to emergency management. *European Journal of Operational Research*, 230(1):113–121, 2013.

[73] Hatice Calik, Martine Labbé, and Hande Yaman. p-Center Problems. In Gilbert Laporte, Stefan Nickel, and Francisco Saldanha da Gama, editors, *Location Science*, pages 79–92. Springer International Publishing, Cham, 2015.

[74] V. Bayram and H. Yaman. A stochastic programming approach for shelter location and evacuation planning. *RAIRO-Operations Research*, 52(3):779–805, 2018.

[75] Isabel Correia and Francisco Saldanha da Gama. Facility location under uncertainty. In *Location science*, pages 177–203. Springer, 2015.

[76] D. Serra and V. Marianov. The p-median problem in a changing network: the case of barcelona. *Location Science*, 6(1-4):383–394, 1998.

[77] L. Snyder and M. S Daskin. Stochastic p-robust location problems. *IIE Transactions*, 38(11):971–985, 2006.

[78] Marc Demange, Marcel A Haddad, and Cécile Murat. Hardness and approximation of the probabilistic p-center problem under pressure. *arXiv preprint arXiv:2009.08751*, 2020.

[79] Marc Demange and Dominique de Werra. On some coloring problems in grids. *Theor. Comput. Sci.*, 472:9–27, 2013.

[80] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.

[81] Marc Demange and Tinaz Ekim. A note on the np-hardness of two matching problems in induced subgrids. *Discrete Mathematics & Theoretical Computer Science*, 15(2):233–242, 2013.

[82] L. Yanpei, A. Morgana, and B. Simeone. General theoretical results on rectilinear embedability of graphs. *Acta Mathematicae Applicatae Sinica*, 7:187–192, 1991.

[83] Mark S. Daskin and Susan Hesse Owen. Two new location covering problems: The partial p-center problem and the partial set covering problem. *Geographical Analysis*, 31(1):217–223, 1999.

[84] Wen-Lian Hsu and George L. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209 – 215, 1979.

[85] Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, 1986.

[86] John E Beasley. Or-library: distributing test problems by electronic mail. *Journal of the operational research society*, 41(11):1069–1072, 1990.

[87] Philip J Fleming and John J Wallace. How not to lie with statistics: the correct way to summarize benchmark results. *Communications of the ACM*, 29(3):218–221, 1986.

[88] Marcel A. Haddad. Rpcp subgrid library. `www.lamsade.dauphine.fr/~mhaddad/Library/`, September 2019.

RÉSUMÉ

A cause du réchauffement climatique, le nombre et l'intensité des feux de forêts augmentent autour du globe. Dans ce contexte, la construction de refuges contre le feu est une solution de plus en plus envisagée. Le problème consiste essentiellement à localiser p refuges de sorte à minimiser la distance maximale qui sépare un usager du plus proche refuge accessible en cas de feux. Le territoire considéré est divisé en zones et est modélisé comme un graphe aux arêtes pondérées. Un départ de feux sur une seule zone (c'est-à-dire sur un sommet). La principale conséquence d'un feu est que les chemins d'évacuation sont modifiés de deux manières. Premièrement, un chemin d'évacuation ne peut pas traverser le sommet en feu. Deuxièmement, le fait qu'une personne proche de l'incendie puisse avoir un choix limité de direction d'évacuation, ou être sous stress, est modélisé à l'aide d'une stratégie d'évacuation nouvellement définie. Cette stratégie d'évacuation induit des distances d'évacuation particulières qui rendent notre modèle spécifique. Selon le type de données considéré et l'objectif recherché, nous proposons deux problèmes avec ce modèle: le *Robust* p*-Center Under Pressure* et le *Probabilistic* p*-Center Under Pressure*. Nous prouvons que ces deux problèmes sont NP-difficiles sur des classes de graphes pertinentes pour notre contexte. Nous proposons également des résultats d'approximation et d'inapproximation. Finalement, nous développons des algorithmes polynomiaux sur des classes de graphes simples, et nous développons des algorithmes mathématiques basés sur la programmation linéaire.

MOTS CLÉS

Gestion des situations d'urgence, Feux de forêt, Localisation des abris sous incertitude, Variantes du problème du p-Centre, Stratégie d'évacuation, Optimisation probabiliste, Optimisation combinatoire robuste, Programmation linéaire en nombres entiers

ABSTRACT

The location of shelters in different areas threatened by wildfires is one of the possible ways to reduce fatalities in a context of an increasing number of catastrophic and severe forest fires. The problem is basically to locate p shelters minimizing the maximum distance people will have to cover to reach the closest accessible shelter in case of fire. The landscape is divided in zones and is modeled as an edge-weighted graph with vertices corresponding to zones and edges corresponding to direct connections between two adjacent zones. Each scenario corresponds to a fire outbreak on a single zone (i.e., on a vertex) with the main consequence of modifying evacuation paths in two ways. First, an evacuation path cannot pass through the vertex on fire. Second, the fact that someone close to the fire may have limited choice, or may not take rational decisions, when selecting a direction to escape is modeled using a new kind of evacuation strategy. This evacuation strategy, called Under Pressure, induces particular evacuation distances which render our model specific. We propose two problems with this model: the *Robust* p*-Center Under Pressure* problem and the *Probabilistic* p*-Center Under Pressure* problem. First we prove hardness results for both problems on relevant classes of graphs for our context. In addition, we propose polynomial exact algorithms on simple classes of graphs and we develop mathematical algorithms based on integer linear programming.

KEYWORDS

Forest fire emergency, Wildfire emergency, Shelter location under uncertainty, Variants of the p-Center problem, Evacuation strategy, Under pressure decision model, Probabilistic Optimization, Robust Combinatorial Optimization, Integer linear programming