Ancrage et robustesse pour le RCPSP: outils exacts et heuristiques

Adèle Pass-Lanneau, Pascale Bendotti, Philippe Chrétienne, Pierre Fouilhoux

Ancrage et robustesse pour le RCPSP : outils exacts et heuristiques

Adèle Pass-Lanneau[1,2], Pascale Bendotti[1,2], Philippe Chrétienne[2], Pierre Fouilhoux[2]

[1] EDF R&D, F-91120 Palaiseau, France
[2] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
adele.pass-lanneau@polytechnique.org

Mots-clés : *RCPSP, optimisation robuste 2-stage, ancrage, PLNE, heuristiques*

1 Optimisation robuste en ordonnancement de projet

En ordonnancement de projet, on considère un ensemble de tâches J auxquelles on doit attribuer une date de début $x_j \geq 0$. Les tâches ont des durées d'exécutions $p \in \mathbb{R}_+^J$, et la date de fin d'exécution maximale est le *makespan* de l'ordonnancement. Un ordonnancement peut être sujet à différentes contraintes. Dans le cas d'une contrainte de précédences entre deux tâches i et j, la tâche j doit commencer après la fin de l'exécution de la tâche i. Dans le cas de contraintes de ressources, chaque tâche consomme une quantité de ressources pendant son exécution, et l'ordonnancement doit être tel que la consommation de ressources des tâches en cours n'excède jamais un seuil prédéterminé. Trouver un ordonnancement de makespan minimal sous contraintes de précédences et de ressources constitue le Resource-Constrained Project Scheduling Problem (RCPSP). Le RCPSP est un problème NP-difficile classique, rencontrant une large variété d'applications pratiques, et qui est dur à résoudre numériquement [1].

Dans un cas réel, les durées d'exécutions des tâches peuvent être des paramètres incertains : soit parce que ces durées sont difficiles à estimer a priori ; soit parce qu'elles risquent de changer suite à une perturbation. Pour prendre en compte ces incertitudes, nous nous intéressons dans ce travail au paradigme de l'optimisation robuste. Les durées des tâches ont des valeurs nominales p_j, mais peuvent être perturbées en $p_j + \delta_j$, avec δ_j une déviation. Le vecteur de déviation δ est supposé être dans un ensemble d'incertitude Δ. Une approche robuste-statique [10] pour le RCPSP serait de déterminer un ordonnancement faisable pour tout $\delta \in \Delta$. Cependant une telle solution aurait un makespan très grand, et serait donc inutilisable. Cela motive l'étude d'autres approches robustes.

Comme souligné dans [7], la littérature en optimisation robuste pour l'ordonnancement de projet est relativement peu abondante, contrairement au pendant stochastique. Dans [6], les auteurs introduisent le RCPSP robuste-ajustable, dérivé du concept général de robustesse ajustable – ou 2-stage – de [3]. Des *décisions de séquencement* sont prises avant la réalisation de l'incertitude, puis l'ordonnancement est décidé après réalisation de l'incertitude, de façon à minimiser le makespan pire cas. Le RCPSP robuste-ajustable permet de garantir un makespan pire cas, mais il ne vise pas à calculer en avance un ordonnancement dit *baseline*. Or, dans un contexte appliqué, précalculer une solution baseline est souvent nécessaire pour préparer la réalisation des tâches. L'*ancrage* a été introduit dans [4] pour l'ordonnancement sous contraintes de précédences seulement. Dans un ordonnancement baseline, un ensemble de tâches est dit *ancré* si les dates de début de ses tâches sont garanties, et restent inchangées quelle que soit la réalisation de l'incertitude.

Dans le présent travail, nous étendons l'ancrage au cas du RCPSP, et formulons un nouveau problème, le RCPSP robuste-ancré. Il vise à calculer un ordonnancement baseline et des décisions de séquencement, de façon à respecter un makespan pire cas fixé, et à maximiser le nombre total de tâches ancrées.

Contributions. Nous définissons le RCPSP robuste-ancré, en lien avec le RCPSP robuste-ajustable. Ces deux problèmes sont complémentaires en pratique pour calculer à la fois un makespan et un ordonnancement robuste. Nous proposons ensuite des outils algorithmiques pour ces deux problèmes de façon unifiée, grâce notamment à une structure de graphe dédiée. Des formulations PLNE compactes sont déduites pour les deux problèmes. Des approches heuristiques sont ensuite proposées. Enfin une évaluation numérique de ces méthodes exactes et heuristiques est menée. Le détail des preuves et résultats peut être trouvé dans [9].

2 Approche robuste-ancrée pour le RCPSP

2.1 Préliminaires : décisions de séquencement

Considérons les notations suivantes pour le RCPSP déterministe. Soit $\bar{J} = J \cup \{s, t\}$ avec s, t deux tâches artificielles représentant le début et la fin du projet. Les relations de précédences sont données par un graphe de précédence $(\bar{J}, \mathcal{A}, p)$ acyclique, de source s et puits t, pondéré sur les arcs par les durées des tâches. L'ensemble des ressources est $\mathcal{R}$. Pour toute ressource $k \in \mathcal{R}$, le seuil de ressource disponible est R_k et une tâche $j \in J$ a une consommation r_{jk} de ressource.

Un outil caractérisant les solutions réalisables pour les contraintes de ressources est le *flot de ressource* introduit par [2]. Il s'agit d'une collection de $s{-}t$ flots $(f^k)_{k\in\mathcal{R}}$ dans le graphe $(\bar{J}, \bar{J} \times \bar{J})$. Le flot f^k a valeur R_k, et la quantité passant par une tâche $i \in J$ est égale à sa consommation de ressource r_{ik}. Intuitivement, le flot f^k représente la façon dont les unités de ressource k passent d'une tâche à l'autre. Si $f^k_{ij} > 0$, alors une contrainte de précédence additionnelle entre i et j doit être satisfaite. Ces contraintes de précédences sont représentées par un vecteur $\sigma \in \{0,1\}^{J^2}$ et on note $\mathcal{A}^\sigma = \{(i,j) \in J^2 :\ \sigma_{ij} = 1\}$. On dit que σ est une *décision de séquencement* si elle est associée à un flot de ressource valide. Soit $\mathfrak{S}$ l'ensemble de toutes les décisions de séquencement. Le RCPSP consiste alors à trouver $\sigma \in \mathfrak{S}$ et un ordonnancement x qui vérifie les contraintes de précédences de $(\bar{J}, \mathcal{A}^\sigma, p)$, de makespan minimal.

Un exemple est donné en Figure 1 sur une instance à 5 tâches. Les arcs pleins sont les arcs de précédence. On considère une seule ressource, avec un seuil $R = 2$, et les consommations des tâches entre crochets. Les arcs gris pointillés représentent un flot de ressource pour cette instance. La décision de séquencement σ associée vérifie $\mathcal{A}^\sigma = \{(1,2),(2,3),(4,5),(4,2),(2,5)\}$.

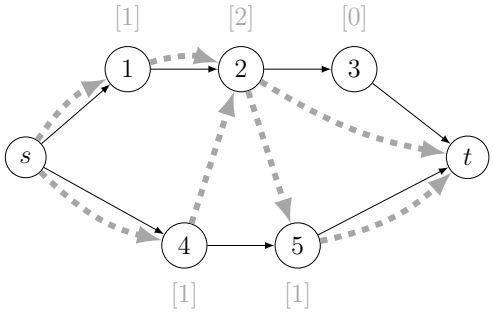

FIG. 1 – Exemple de graphe de précédence et d'un flot de ressource.

2.2 Le RCPSP robuste-ancré

Nous considérons le processus de décision 2-stage suivant. Une solution baseline (x, σ) est choisie en première étape, réalisable pour l'instance nominale du problème $\mathcal{I}$. Puis, l'instance réelle s'avère être $\mathcal{I}^\delta$: l'ordonnancement peut être modifié en deuxième étape en un nouvel ordonnancement x^δ de $(\bar{J}, \mathcal{A}^\sigma, p{+}\delta)$, avec la décision de séquencement σ inchangée. Le décideur peut également vouloir garantir les dates des tâches, pour qu'elles soient inchangées entre x et x^δ. Pour formaliser cette idée, nous définissons les tâches ancrées, en étendant la définition donnée dans [4].

Définition 1 *Soit (z, σ) une solution de l'instance $\mathcal{I}$ du RCPSP. Soit $\overline{H} \subseteq \bar{J}$. L'ensemble $\overline{H}$ est* ancré *par rapport à l'ordonnancement z et la décision de séquencement σ si pour tout $\delta \in \Delta$, il existe un ordonnancement z^δ de $(\bar{J}, \mathcal{A}^\sigma, p+\delta)$ tel que $z_i = z_i^\delta$ pour tout $i \in \overline{H}$.*

Les tâches de $\overline{H}$ sont ainsi "à dates garanties" puisqu'il est toujours possible de réparer l'ordonnancement baseline sans modifier leurs dates de début. Dans la suite, nous appelons *solutions ancrées* les triplets $(z, \sigma, \overline{H})$, avec $\overline{H}$ ancré par rapport à z et σ.

Le problème **RCPSP robuste-ancré** peut désormais être défini comme suit. Etant donné une instance $\mathcal{I} = ((\bar{J}, \mathcal{A}, p), (r_{ik})_{i \in J, k \in \mathcal{R}}, (R_k)_{k \in \mathcal{R}})$ du RCPSP et une deadline $M \geq 0$, trouver une décision de séquencement $\sigma \in \mathfrak{S}$, un ordonnancement baseline z associé, et un sous-ensemble de tâches $H \subseteq J$, tels que le makespan de z est au plus M, l'ensemble $H \cup \{t\}$ est ancré par rapport à z et σ, et le nombre de tâches ancrées $|H|$ est maximal.

C'est un problème robuste 2-stage qui peut être réécrit sous la forme suivante :

$$\begin{array}{lll} \max & \min & \max \quad |H| \\ \sigma \in \mathfrak{S} & \delta \in \Delta & z^\delta \text{ ordonnancement de } (\bar{J}, \mathcal{A}^\sigma, p+\delta) \\ z \text{ ordonnancement de } (\bar{J}, \mathcal{A}^\sigma, p) & & z_i^\delta = z_i \ \forall i \in H \\ z_t \leq M & & z_t^\delta = z_t \\ H \subseteq J & & \end{array}$$

A titre d'exemple, considérons à nouveau l'instance de la Figure 1. Les tâches sont de durée unitaire. Dans la Figure 2 est représenté le diagramme de Gantt d'un ordonnancement réalisable $z = (0, 0, 2, 3, 1, 3, 5)$. Chaque tâche peut être allongée, les allongements maximaux étant $(1, 1, 1, 0.5, 0.5)$ représentés par les rectangles hachurés. On considère les réalisations de l'incertitude où au plus une tâche peut être allongée. La Figure 2 représente alors une solution ancrée formée d'un ordonnancement baseline vérifiant la deadline $M = 5$, des décisions de séquencement σ issues de la Figure 1, et de tâches ancrées $\{1, 3, 5, t\}$. Quelle que soit la tâche allongée, on peut modifier l'ordonnancement baseline en un ordonnancement réalisable en réordonnançant seulement les tâches 2 et 4, sans modifier les décisions de séquencement, et tout en respectant la deadline $M = 5$.

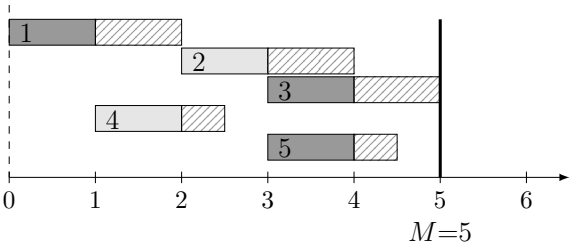

FIG. 2 – Exemple d'une solution ancrée pour la deadline $M = 5$, tâches ancrées en gris foncé.

2.3 Lien avec le RCPSP robuste-ajustable

Le RCPSP robuste-ajustable défini dans [6] s'écrit comme

$$\begin{array}{llll} \min & \max & \min & x_t^\delta \\ \sigma \in \mathfrak{S} & \delta \in \Delta & \text{s.t.} & x^\delta \text{ ordonnancement de } (\bar{J}, \mathcal{A}^\sigma, p+\delta) \end{array}$$

où le problème max-min intérieur correspond au makespan pire cas associé à σ. Le lien entre le RCPSP robuste-ajustable et les solutions ancrées est basé sur l'observation suivante. On peut vérifier que le makespan pire cas associé à une décision de séquencement σ est égal au makespan minimum d'un ordonnancement z, où $(z, \sigma, \{t\})$ une solution ancrée. Intuitivement, le problème ajustable vise à garantir la date de la tâche fictive finale t, c'est-à-dire à l'ancrer.

Le RCPSP robuste-ajustable et le RCPSP robuste-ancré peuvent être utilisés de manière combinée : tout d'abord, trouver un makespan pire cas M en utilisant le RCPSP robuste-ajustable ; ensuite, trouver une solution baseline avec des tâches ancrées respectant cette deadline M en utilisant le RCPSP robuste-ancré.

3 Outils de résolution exacte et approchée

Soit Δ l'ensemble à budget [5] défini par une déviation max $\widehat{\delta} \in \mathbb{R}_+^J$ et un budget Γ : $\Delta = \{\delta = (\widehat{\delta}_i u_i)_{i\in J} \colon\ u \in \{0,1\}^J,\ \sum_{i\in J} u_i \leq \Gamma\}$. Afin de développer des outils de résolution pour le RCPSP robuste-ancré sous budget d'incertitude, nous proposons tout d'abord une représentation par un graphe dédié dit graphe en couches, qui généralise [4].

3.1 Graphe en couches

Pour $\overline{H} \subseteq \bar{J}$, le *graphe en couches* $G^{\text{lay}}(\mathcal{A}, \overline{H})$ est défini de la façon suivante. Il contient $\Gamma+1$ *couches*, indicées de 0 à Γ. Chaque couche γ contient une copie de l'ensemble des tâches, notée i^γ, $i \in \bar{J}$. Les arcs du graphe en couches sont de trois types. Les *arcs horizontaux* sont les arcs (i^γ, j^γ) pour tout γ et $(i,j) \in \mathcal{A}$, avec poids p_i. Les *arcs transversaux* sont les arcs $(i^{\gamma+1}, j^\gamma)$ pour tout $\gamma < \Gamma$ et $(i,j) \in \mathcal{A}$, avec poids $p_i + \widehat{\delta}_i$. Enfin les *arcs verticaux* sont les arcs (j^γ, j^Γ) pour tout $j \in \overline{H}$, $\gamma < \Gamma$, avec poids 0.

Nous obtenons la caractérisation suivante des solutions ancrées, via le graphe en couches.

Théorème 1 *Soit (z,σ) une solution d'une instance $\mathcal{I}$ du RCPSP. Soit $\overline{H} \subseteq \bar{J}$. L'ensemble $\overline{H}$ est ancré par rapport à σ et z si et seulement si il existe x un ordonnancement de $G^{\text{lay}}(\mathcal{A}^\sigma, \overline{H})$ tel que $x_i^\Gamma = z_i$ pour tout $i \in \bar{J}$.*

Ainsi, la définition d'un ensemble ancré, impliquant un problème min/max, peut être remplacée par la simple existence d'un ordonnancement du graphe en couches.

Une conséquence du théorème est que pour tout $\sigma \in \mathfrak{S}$, le makespan pire cas associé à σ est égal à la valeur minimale de x_t^Γ pour x un ordonnancement de $G^{\text{lay}}(\mathcal{A}^\sigma, \{t\})$. Autrement dit, la valeur robuste-ajustable associée à une décision de séquencement σ se calcule simplement par un parcours du graphe en couches.

3.2 Reformulations PLNE

Pour le problème RCPSP robuste-ancré, nous obtenons une formulation composée :
– de variables f et σ comme dans la formulation flot du RCPSP déterministe de [2] ;
– de variables continues $x_j^\gamma \geq 0$ pour tout $j \in \bar{J}$, $\gamma \in \{0,\dots,\Gamma\}$, qui encodent un ordonnancement du graphe en couches ;
– de variables binaires indiquant les tâches ancrées $h \in \{0,1\}^J$.

Théorème 2 *Le RCPSP robuste-ancré sous budget d'incertitude Γ admet la reformulation PLNE suivante, notée (F_{anch}).*

$$
\begin{array}{lll}
(F_{\text{anch}}) & \max & \sum_{i\in J} h_i \\
& \text{s.t.} & x_j^\gamma - x_i^\gamma \geq p_i - M(1-\sigma_{ij}) \qquad \forall i,j \in J^2,\ \forall \gamma \leq \Gamma \qquad (1)\\
& & x_j^\gamma - x_i^{\gamma+1} \geq p_i + \widehat{\delta}_i - M(1-\sigma_{ij}) \qquad \forall i,j \in J^2,\ \forall \gamma < \Gamma \qquad (2)\\
& & x_t^\Gamma - x_t^\gamma \geq 0 \qquad \forall \gamma < \Gamma \qquad (3)\\
& & x_j^\Gamma - x_j^\gamma \geq -M(1-h_j) \qquad \forall j \in J,\ \forall \gamma < \Gamma \qquad (4)\\
& & x_t^\Gamma \leq M \qquad (5)\\
& & \sigma,\ f \text{ décision de séquencement et flot associé} \qquad (6)\\
& & f_{ij}^k \geq 0 \qquad \forall i,j \in J^2, k \in \mathcal{R}\\
& & \sigma_{ij} \in \{0,1\} \qquad \forall i,j \in J^2\\
& & x_j^\gamma \geq 0 \qquad \forall j \in \bar{J},\ \forall \gamma \leq \Gamma\\
& & h_j \in \{0,1\} \qquad \forall j \in J
\end{array}
$$

Les contraintes (1-4) sont les contraintes de précédences du graphe en couche, où la deadline M joue le rôle de bigM. La contrainte (5) est la contrainte de deadline. Les contraintes (6) sont issues de la formulation flot de [2].

Une formulation similaire pour le RCPSP robuste-ajustable, notée (F_{adj}), est également obtenue. Notons qu'il s'agit de reformulations PLNE exactes et compactes, ce qui est rare dans la littérature pour des problèmes robustes 2-stage.

3.3 Heuristiques

Etant donné la difficulté du RCPSP déterministe, les problèmes RCPSP robuste-ancré et robuste-ajustable sont a priori au moins aussi difficiles à résoudre numériquement. Cela motive le développement d'heuristiques pour résoudre ces deux problèmes.

Nous proposons les approches suivantes.
– **RCPSP robuste-ajustable.** Nous utilisons des heuristiques classiques basées sur ParallelSGS et des règles de priorité [8] pour le RCPSP déterministe, qui permettent de calculer des décisions de séquencement. Nous montrons que l'usage de ces heuristiques peut être combiné à un parcours du graphe en couches, afin d'obtenir un algorithme polynomial noté ($\mathcal{H}_{\text{adj}}$) produisant une solution heuristique du RCPSP robuste-ajustable.
– **RCPSP robuste-ancré.** Nous supposons que la deadline M a été obtenue en résolvant le problème RCPSP robuste-ajustable précédemment. De ce fait, une décision de séquencement initiale est connue. Nous proposons dans ce cas une matheuristique ($\mathcal{H}_{\text{anch}}$) qui consiste à résoudre le PLNE (F_{anch}) "à flot fixé", en conservant la décision de séquencement initiale.

4 Résultats numériques

Nous évaluons nos approches PLNE et heuristiques pour le RCPSP robuste-ajustable et le RCPSP robuste-ancré. Le benchmark utilisé est celui de [6], il est basé sur les instances PSPLib avec un nombre de tâches $n \in \{30, 60, 90, 120\}$, auxquelles on ajoute des déviations et un budget d'incertitude $\Gamma \in \{3, 5, 7\}$. Les formulations ont été implémentées avec Julia 0.6.2 et résolues avec CPLEX 12.8, avec une limite en temps de 1200 secondes.

Les principales conclusions numériques sont les suivantes.
– **RCPSP robuste-ajustable.** Sur les instances à 30 tâches, la reformulation PLNE exacte (F_{adj}) résout à l'optimum 1026 instances sur 1440 ; soit 259 de plus que la référence de la littérature [6] basée sur une décomposition de Benders. Nous identifions que les instances les plus difficiles sont celles qui sont difficiles à résoudre dans le cas déterministe avec la formulation flot. Le surcoût computationnel du robuste, par rapport à la formulation flot pour le problème déterministe, est faible. Sur les instances plus grandes, nous analysons l'impact des paramètres du benchmark sur la qualité de la résolution. Notons que les instances à 90 tâches sont majoritairement non résolues. L'heuristique ($\mathcal{H}_{\text{adj}}$) permet de trouver des solutions en temps très court, inférieur à 1 seconde. La qualité de cette solution est évaluée. Sur les instances à 30 tâches dont on connaît la solution optimale, l'heuristique trouve l'optimum pour 596 instances sur 1018. Sur certaines instances à 90 tâches non résolues par PLNE, l'heuristique trouve des solutions meilleures que la meilleure solution trouvée par PLNE après 1200 secondes.

– **RCPSP robuste-ancré.** La formulation PLNE (F_{anch}) permet d'obtenir le nombre optimal de tâches ancrées sur les instances considérées. Sur les instances à 30 tâches, on obtient ainsi 19 tâches ancrées en moyenne. Ainsi, sans augmenter le makespan par rapport à ce qui a été calculé par le problème robuste-ajustable, on peut trouver un ordonnancement avec un nombre significatif de tâches à dates garanties. Concernant la qualité de résolution, nous observons que les difficultés observées pour le problème robuste-ajustable sont également présentes pour le problème robuste-ancré. La matheuristique ($\mathcal{H}_{\text{anch}}$) proposée permet en revanche d'obtenir des solutions en temps court, inférieur à 1 seconde en moyenne sur toutes les tailles d'instances du benchmark.

5 Conclusion

Dans ce travail, nous proposons le concept de solutions ancrées pour le RCPSP. Le problème RCPSP robuste-ancré permet d'obtenir des garanties sur les dates des tâches dans un ordonnancement baseline, en plus de garantir le makespan pire cas.

Les approches exactes et heuristiques proposées sont basées sur une étude dédiée du problème sous budget d'incertitude, en utilisant le flot de ressource proposé dans la littérature. L'ensemble des approches proposées forme une boîte à outils complète pour la résolution de ces problèmes.

L'évaluation numérique de nos méthodes montre la difficulté des deux problèmes considérés, qui sont plus durs que le RCPSP. Elle montre cependant que le surcoût de calcul lié à l'aspect robuste est faible. Ainsi, de bons outils algorithmiques pour le RCPSP déterministe donnent de bons outils algorithmiques pour le RCPSP robuste-ancré ou robuste-ajustable.

Une perspective est d'améliorer la résolution de ces problèmes, par exemple par le renforcement des formulations PLNE établies, ou en utilisant les heuristiques à base de règles de priorité au cours de la résolution. Enfin, une extension serait de considérer d'autres problèmes robustes où cette fois les décisions de séquencement pourraient être revues en deuxième étape.

Références

[1] Christian Artigues, Sophie Demassey, and Emmanuel Neron. *Resource-Constrained Project Scheduling : Models, Algorithms, Extensions and Applications*. ISTE/Wiley, 2008.

[2] Christian Artigues, Philippe Michelon, and Stéphane Reusser. Insertion techniques for static and dynamic resource-constrained project scheduling. *Eur. J. Oper. Res.*, 149(2) :249–267, 2003.

[3] Aharon Ben-Tal, A. P. Goryashko, E. Guslitzer, and Arkadi Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2) :351–376, 2004.

[4] Pascale Bendotti, Philippe Chrétienne, Pierre Fouilhoux, and Adèle Pass-Lanneau. The Anchor-Robust Project Scheduling Problem. https ://hal.archives-ouvertes.fr/hal-02144834. May 2019.

[5] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52 :35–53, 2004.

[6] Maria Bruni, Luigi Di Puglia Pugliese, Patrizia Beraldi, and Francesca Guerriero. An adjustable robust optimization model for the resource-constrained project scheduling problem with uncertain activity durations. *Omega*, 09 2016.

[7] Öncü Hazır and Gündüz Ulusoy. A classification and review of approaches and methods for modeling uncertainty in projects. *International Journal of Production Economics*, 223 :107522, 2020.

[8] Rainer Kolisch and Sönke Hartmann. *Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem : Classification and Computational Analysis*, pages 147–178. Springer US, Boston, MA, 1999.

[9] Adèle Pass-Lanneau. *Anchored solutions in robust combinatorial optimization*. PhD thesis, 2021. Thèse de doctorat dirigée par Fouilhoux, Pierre et Bendotti, Pascale. Informatique, Sorbonne université 2021.

[10] Allen L. Soyster. Technical note – convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5) :1154–1157, 1973.