PROJET MOGPL : OPTIMISATION EQUITABLE

Ben KABONGO

Nour BOUCHOUCHI

4 décembre 2022

1 Linéarisation de f

1.1 Montrons que $L_k(z)$ est la solution optimale de $P(k)$

On sait que la somme $\sum_{i=1}^{k} a_{ik}$ vaut k avec $a_{ik} \in \{0, 1\}$. Alors, minimiser cette somme revient à sélectionner les k plus petits z_i.

Etant donné que les $z_{(i)}$ représentent le tri par ordre croissant des composantes z_i, il faut donc sélectionner les k premiers $z_{(i)}$.

Or, par définition, $L_k(z) = \sum_{i=1}^{k} z_{(i)}$, et les $z_{(i)}$ sont triés par ordre croissant.

La valeur de la fonction objectif à l'optimum est, par conséquent, égale à :

$$\sum_{i=1}^{k} z_{(i)} = L_k(z).$$

1.2 Le dual de $P(k)$ est :

$$\max\ kr_k - \sum_{i=1}^{n} b_{ik}$$

$$\text{s.c.} \left\{ \begin{array}{l} r_k - b_{ik} \leq z_i \quad i = 1, ..., n \quad k = 1, ..., n \end{array} \right.$$

$$r_k \in \mathbb{R},\ b_{ik} \geq 0,\ i = 1, ..., n,\ k = 1, ..., n$$

Appliqué à $z = (4, 7, 1, 3, 9, 2)$ et en faisant varier k de 1 à 6, nous obtenons par programmation linéaire : $L(z) = (1, 3, 6, 10, 17, 26)$.

Ce résultat correspond bien au résultat attendu. En effet, une fois trié par ordre croissant on trouve que $z = (1, 2, 3, 4, 7, 9)$ et donc : $L(z) = (1, 1+2, 1+2+3, 1+2+3+4, 1+2+3+4+7, 1+2+3+4+7+9) = (1, 3, 6, 10, 17, 26)$.

1.3 Montrons que $f(x) = \sum_{k=1}^{n} w'_k L_k(z(x))$

$$f(x) = \sum_{k=1}^{n} w_k z_{(k)}$$

$$f(x) = w_1 z_{(1)} + \sum_{k=2}^{n} w_k(z_{(k)})$$

$$f(x) = w_1 z_{(1)} + \sum_{k=2}^{n} w_k(L_k(z(x)) - L_{k-1}(z(x)))$$

$$f(x) = w_1 L_1(z(x)) + \sum_{k=2}^{n} w_k L_k(z(x)) - w_k L_{k-1}(z(x))$$

$$f(x) = \sum_{k=1}^{n} w_k L_k(z(x)) - \sum_{k=2}^{n} w_k L_{k-1}(z(x))$$

$$f(x) = w_n z_{(n)} + \sum_{k=1}^{n-1} w_k L_k(z(x)) - \sum_{k=1}^{n-1} w_{k+1} L_k(z(x))$$

$$f(x) = w'_n z_{(n)} + \sum_{k=1}^{n-1} (w_k - w_{k+1}) L_k(z(x))$$

$$f(x) = w'_n z_{(n)} + \sum_{k=1}^{n-1} w'_k L_k(z(x))$$

Donc $f(x) = \sum_{k=1}^{n} w'_k L_k(z(x))$

1.4 Linéarisation de f et programme linéaire de l'exemple 1

On se place dans un problème dont l'objectif serait de sélectionner un ensemble de trois objets parmi six de manière à satisfaire équitablement deux agents en sachant que les utilités des objets selon les deux agents sont respectivement $(5, 6, 4, 8, 1)$ et $(3, 8, 6, 2, 5)$.

Dans le cas où le vecteur de pondération est $w = (2, 1)$, le programme linéaire correspondant est le suivant :

$$\max r_1 - b_{11} - b_{21} + 2r_2 - b_{12} - b_{22}$$

$$\text{s.c.} \begin{cases} z_1 = 5x_1 + 6x_2 + 4x_3 + 8x_4 + x_5 \\ z_2 = 3x_1 + 8x_2 + 6x_3 + 2x_4 + 5x_5 \\ x_1 + x_2 + x_3 + x_4 + x_5 = 3 \\ r_1 - b_{11} \leq z_1 \\ r_1 - b_{21} \leq z_2 \\ r_2 - b_{12} \leq z_1 \\ r_2 - b_{22} \leq z_2 \end{cases}$$

$$r_k \in \mathbb{R},\ x_j \in \{0,1\},\ b_{ik} \geq 0,\ i = 1,2,\ k = 1,2,\ j = 1,...,5$$

La solution optimale est donnée par : $x_1 = 0$, $x_2 = 1$, $x_3 = 1$, $x_4 = 1$, $x_5 = 0$. Cela revient donc à sélectionner le deuxième, le troisième et le quatrième objet.

La valeur de la fonction objectif à l'optimum est alors de 50, avec $z_1 = 18$ et $z_2 = 16$ qui correspondent respectivement à la satisfaction du premier et du second individu.

2 Application au partage équitable de biens indivisibles

2.1 Linéarisation

Soit c_j le coût de l'objet j pour tous les agents.

Notons x_{ij} la variable booléenne qui vaut 1 si l'objet j est attribué à l'agent i.

Le programme correspondant au partage s'écrit :

$$\max f(x)$$

$$\text{s.c.} \begin{cases} z_i = \sum_{j=1}^{p} c_j x_{ij} \quad i = 1, ..., n \\ \sum_{i=1}^{n} x_{ij} \leq 1 \quad j = 1, ..., p \end{cases}$$

$$x_{ij} \in \{0, 1\}, \ i = 1, ..., n, \ j = 1, ..., p$$

Après linéarisation de f, on obtient le programme linéaire suivant :

$$\max \sum_{k=1}^{n} w'_k (k r_k - \sum_{i=1}^{n} b_{ik})$$

$$\text{s.c.} \begin{cases} z_i = \sum_{j=1}^{p} c_j x_{ij} \quad i = 1, ..., n \\ \sum_{i=1}^{n} x_{ij} \leq 1 \quad j = 1, ..., p \\ r_k - b_{ik} \leq z_i \quad i = 1, ..., n \quad k = 1, ..., n \end{cases}$$

$$r_k \in \mathbb{R}, \ x_{ij} \in \{0, 1\}, \ b_{ik} \geq 0, \ i = 1, ..., n, \ k = 1, ..., n, \ j = 1, ..., p$$

On considère un problème de partage équitable de six objets de valeurs respectives 325, 225, 210, 115, 75 et 50 euros entre trois agents A, B et C. Dans ce cas particulier, l'utilité d'un objet j est la même pour les trois agents. Elle est égale à la valeur de l'objet j.

Appliqué ce problème, on obtient le programme linéaire suivant pour $w = (10, 3, 1)$:

$$\max\ 7r_1 - 7b_{11} - 7b_{21} - 7b_{31} + 4r_2 - 2b_{12} - 2b_{22} - 2b_{32} + 3r_3 - b_{13} - b_{23} - b_{33}$$

$$\text{s.c.} \begin{cases} z_i = 325x_{i1} + 225x_{i2} + 210x_{i3} + 115x_{i4} + 75x_{i5} + 50x_{i6} \quad i = 1,2,3 \\ \sum_{i=1}^{3} x_{ij} \leq 1 \quad j = 1, ..., 6 \\ r_k - b_{ik} \leq z_i \quad i = 1,2,3, \quad k = 1,2,3 \end{cases}$$

$$r_k \in \mathbb{R},\ x_{ij} \in \{0,1\},\ b_{ik} \geq 0,\ i = 1,2,3,\ k = 1,2,3,\ j = 1, ..., 6$$

On notera également que l'on pourrait rajouter une contrainte pour borner la quantité maximale attribuable aux agents.

Solutions

1. $w = (10, 3, 1)$: $z_1 = 340$, $z_2 = 335$, $z_3 = 325$.

$x_{11} = 0$	$x_{21} = 0$	$x_{31} = 1$
$x_{12} = 1$	$x_{22} = 0$	$x_{32} = 0$
$x_{13} = 0$	$x_{23} = 1$	$x_{33} = 0$
$x_{14} = 1$	$x_{24} = 0$	$x_{34} = 0$
$x_{15} = 0$	$x_{25} = 1$	$x_{35} = 0$
$x_{16} = 0$	$x_{26} = 1$	$x_{36} = 0$

2. $w = (3, 2, 1)$: $z_1 = 340$, $z_2 = 335$, $z_3 = 325$.

$x_{11} = 0$	$x_{21} = 0$	$x_{31} = 1$
$x_{12} = 1$	$x_{22} = 0$	$x_{32} = 0$
$x_{13} = 0$	$x_{23} = 1$	$x_{33} = 0$
$x_{14} = 1$	$x_{24} = 0$	$x_{34} = 0$
$x_{15} = 0$	$x_{25} = 1$	$x_{35} = 0$
$x_{16} = 0$	$x_{26} = 1$	$x_{36} = 0$

3. Moyenne : $w = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$: $z_1 = 1000$, $z_2 = 0$, $z_3 = 0$.

$$\begin{array}{lll}
x_{11} = 1 & x_{21} = 0 & x_{31} = 0 \\
x_{12} = 1 & x_{22} = 0 & x_{32} = 0 \\
x_{13} = 1 & x_{23} = 0 & x_{33} = 0 \\
x_{14} = 1 & x_{24} = 0 & x_{34} = 0 \\
x_{15} = 1 & x_{25} = 0 & x_{35} = 0 \\
x_{16} = 1 & x_{26} = 0 & x_{36} = 0
\end{array}$$

On observe tout d'abord qu'à l'optimum les valeurs des z_i données par les vecteurs de pondération $w = (10, 3, 1)$ et $w = (3, 2, 1)$ correspondent tous deux à l'affectation des objets 2 et 4 à l'agent A, des objet 3, 5 et 6 à l'agent B et de l'objet 1 à l'agent C. Ainsi, les trois agents obtiennent finalement un lot d'objets valant respectivement 340, 335 et 325 euros.

Maximiser la satisfaction moyenne des individus définie par $\frac{(z_1(x)+z_2(x)+z_3(x))}{3}$ revient à choisir un vecteur w de composantes égales. Pour ce cas particulier, on notera que la fonction objectif étant déjà linéaire, on pourrait ne pas la linéariser. Dans ce cas, tous les objets sont attribués à l'individu A. Cette solution permet en effet de maximiser la satisfaction moyenne puisque la totalité des objets ont bien été attribués mais ne prend pas en compte le principe d'équité.

Le vecteur L des deux premières solutions est donné par $L1 = L2 = L(325, 335, 340) = (325, 660, 1000)$.

Le vecteur L de la troisième solution est donné par $L3 = L(0, 0, 1000) = (0, 0, 1000)$.

$L1$ domine $L3$ au sens de Pareto. Par conséquent, les deux premiers partages sont plus équitables que le partage qui maximise la satisfaction moyenne des agents.

2.2 Etude du temps de résolution en fonction de n

On souhaite désormais étudier l'évolution du temps de résolution d'un tel programme linéaire en fonction du nombre d'individus n.

Ayant une licence bridée de gurobi nous n'avons pas pu résoudre le programme linéaire pour $n = 20$ en raison du trop grand nombre de variables et de contraintes de celui-ci. Nous avons donc moyenné le temps de 10 résolutions pour $n = 4, 8, 12, 16$.

On obtient alors le graphe ci-dessous :

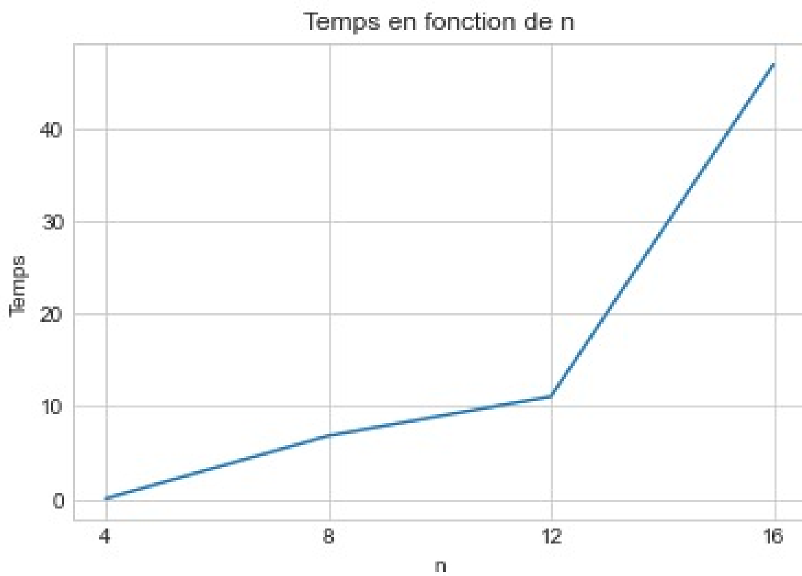


FIGURE 1 – Evolution du temps de résolution en fonction de n

On observe que le temps d'exécution augmente de façon exponentielle lorsque n augmente. Ce résultat correspond à l'intuition que l'on pouvait avoir. En effet, pour n individus le programme linéaire comporte $6n^2 + n$ variables (n variables r_k, n^2 variables b_{ik} et $np = 5n^2$ variables x_{ij} puisque l'on a alors $p = 5n$ objets) et $n^2 + 5n$ contraintes. Pour $n = 4$, le programme comporte donc 100 variables et 36 contraintes. Lorsque $n = 16$ ces nombres augmentent à 1552 et 336 respectivement.

3 Application à la sélection multicritère de projets

3.1 Linéarisation

Soient u_{ij} l'utilité du projet j pour l'agent i et c_j le coût du projet j.

Notons x_j la variable booléenne qui vaut 1 si le projet j est sélectionné.

Le programme correspondant à la sélection multicritère de projets s'écrit :

$$\max f(x)$$

$$\text{s.c.} \begin{cases} z_i = \displaystyle\sum_{j=1}^{p} u_{ij} x_j \quad i = 1, ..., n \\ \displaystyle\sum_{j=1}^{p} c_j x_j \leq b \end{cases}$$

$$x_j \in \{0, 1\},\ j = 1, ..., p$$

Après linéarisation de f, on a :

$$\max \sum_{k=1}^{n} w'_k (k r_k - \sum_{i=1}^{n} b_{ik})$$

$$\text{s.c.} \begin{cases} z_i = \displaystyle\sum_{j=1}^{p} u_{ij} x_j \quad i = 1, ..., n \\ \displaystyle\sum_{j=1}^{p} c_j x_j \leq b \\ r_k - b_{ik} \leq z_i \quad i = 1, ..., n \quad k = 1, ..., n \end{cases}$$

$$r_k \in \mathbb{R},\ x_j \in \{0, 1\},\ b_{ik} \geq 0,\ i = 1, ..., n,\ k = 1, ..., n,\ j = 1, ..., p$$

On s'intéresse désormais à une sélection multicritère de projets. Le but est de sélectionner un ensemble de projets parmi 4 projets dont les coûts sont respectivement de 40, 50, 60 et 50 milliers d'euros en étant contraint par un budget de 100 mille euros. Les utilités de chaque projet selon deux individus sont respectivement $(19, 6, 17, 2)$ et $(2, 11, 4, 18)$.

Appliqué à ce problème, on obtient le programme linéaire suivant pour $w = (10, 1)$:

$$\max\ 9r_1 - 9b_{11} - 9b_{21} + 2r_2 - b_{12} - b_{22}$$

$$\text{s.c.} \begin{cases} z_1 = 19x_1 + 6x_2 + 17x_3 + 2x_4 \\ z_2 = 2x_1 + 11x_2 + 4x_3 + 18x_4 \\ 40x_1 + 50x_2 + 60x_3 + 50x_4 \leq 100 \\ r_k - b_{ik} \leq z_i \quad i = 1,2 \quad k = 1,2 \end{cases}$$

$$r_k \in \mathbb{R},\ x_j \in \{0,1\},\ b_{ik} \geq 0,\ i = 1,2,\ k = 1,2,\ j = 1,...,4$$

Solutions

1. $w = (2, 1)$:

   La solution optimale donne $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, $x_4 = 1$. Cela revient à sélectionner le premier et le dernier projet.

   A l'optimum, $z_1 = 21$ et $z_2 = 20$.

   La valeur de la fonction objectif est 61.

2. $w = (10, 1)$:

   La solution optimale donne $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, $x_4 = 1$. Cela revient également à sélectionner le premier et le dernier projet.

   A l'optimum, $z_1 = 21$ et $z_2 = 20$.

   La valeur de la fonction objectif est 221. La solution est la même que en (1) et ne diffère que par la valeur de la fonction objectif.

3. Moyenne : $w = (1/2, 1/2)$:

   La solution optimale donne $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, $x_4 = 0$. Cela revient à sélectionner le premier et le troisième projet.

   A l'optimum, $z_1 = 36$ et $z_2 = 6$.

   La valeur de la fonction objectif est 21.

A l'optimum, le vecteur d'utilité $(z_1, z_2) = (21, 20)$ est bien équilibré. En revanche, le vecteur d'utilité $(z_1, z_2) = (36, 6)$ dénote d'une répartition très inéquitable des satisfactions.

3.2 Etude du temps de résolution en fonction de n et p

On souhaite désormais étudier l'évolution du temps de résolution d'un tel programme linéaire en fonction du nombre d'individus n et du nombre de projets à sélectionner p. En moyennant le temps de dix résolutions pour $n = 2, 5, 10$ et $p = 5, 10, 15, 20$, on obtient le graphique suivant présentant le temps de résolution du programme linéaire en fonction de n et p.

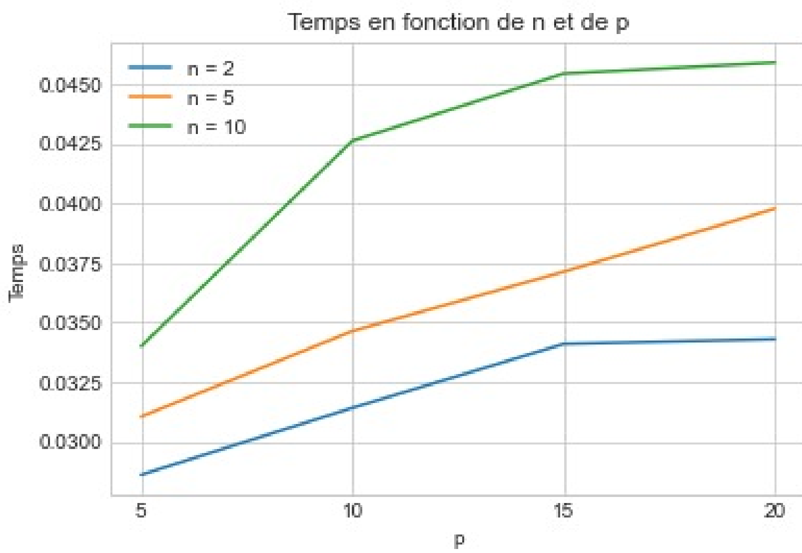


FIGURE 2 – Evolution du temps de résolution en fonction de n et p

On observe ici que le temps de résolution a tendance à augmenter lorsque n augmente. On observe également que lorsque le nombre de projets p à sélectionner augmente le temps de résolution augmente également de façon linéaire. Néanmoins, le nombre d'individus semble avoir une incidence plus importante sur le temps de résolution que le nombre de projets à sélectionner. Cela s'explique par le nombre croissant de variables $n^2 + n + p$ et de contraintes $n^2 + 1$ qui dépend plus de n que de p.

4 Application à la recherche d'un chemin robuste dans un graphe

4.1 Chemin le plus rapide

Soit $G = (V, E)$, un graphe orienté, avec V l'ensemble des sommets et E l'ensemble des arcs.

Soit $s \in S$, un scénario, dont on connait pour chaque arc $(i, j) \in E$, le coût de déplacement de i à j donné par t^s_{ij}.

Soient $a \in V$ et $g \in V$, respectivement un sommet de départ et un sommet de destination.

Notons x_{ij} la variable binaire qui vaut 1 si l'arc (i, j) fait parti du chemin le plus rapide de a à g.

Le programme linéaire de recherche du chemin le plus rapide de a à g s'écrit :

$$\min \sum_{(i,j)\in E} t^s_{ij} x_{ij}$$

$$\text{s.c.} \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} +1 \quad si \quad i = a \\ -1 \quad si \quad i = g \\ O \quad sinon \end{cases} \text{, pour tout } i \in V$$

$$x_{mn} \in \{0, 1\},\ (m, n) \in E$$

Solutions

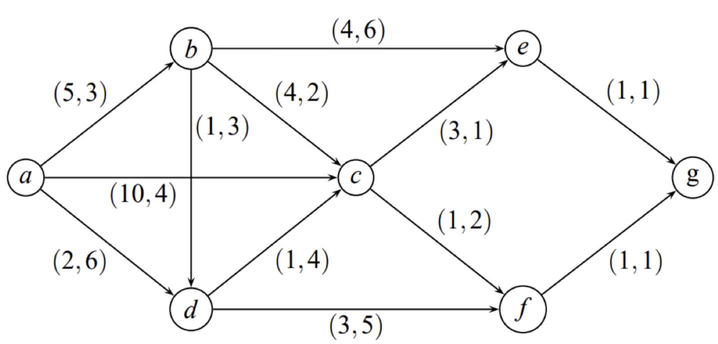


FIGURE 3 – Instance du problème de chemin robuste à 2 scénarios

Soit la liste des arcs de G triés d'après l'ordre suivant :

$$\{(a,b),(a,c),(a,d),(b,d),(b,e),(b,c),(c,d),(c,e),(d,f),(c,f),(e,g),(f,g)\}.$$

Soient leurs coûts respectifs donnés par les deux scénarios :

1. $s1 = (5, 10, 2, 1, 4, 4, 1, 3, 3, 1, 1, 1)$
2. $s2 = (3, 4, 6, 3, 6, 2, 4, 1, 5, 2, 1, 1)$.

La résolution des programmes linéaires pour les deux scénarios donne :

1. Pour le premier scénario on trouve que $x_{ad} = x_{dc} = x_{cf} = x_{fg} = 1$, les autres x_{ij} étant à 0.

   On en déduit que le chemin le plus rapide dans ce scénario consiste à emprunter les arcs (a,d), (d,c), (c,f) et (f,g).

   Ce chemin est donc (a,d,c,f,g).

   Le temps correspondant au parcours de ce chemin est donné par la valeur de la fonction objectif à l'optimum qui vaut ici 5.

2. Dans le second scénario on trouve cette fois que $x_{ac} = x_{ce} = x_{eg} = 1$, les autres x_{ij} valent 0.

   On en déduit que le chemin le plus rapide dans ce scénario consiste à emprunter les arcs (a,c), (c,e) et (e,g).

   Ce chemin est donc (a,c,e,g).

   Le temps correspondant au parcours de ce chemin est donné par la valeur de la fonction objectif à l'optimum qui vaut ici 6.

4.2 Chemin robuste en fonction des deux scénarios

On souhaite désormais trouver un chemin robuste allant de a à g. Celui-ci doit donc être rapide dans les deux scénarios.

Un tel programme peut s'écrire :

$$\max v_f(P) = f(-t^1(P), -t^2(P))$$

$$\text{s.c.} \begin{cases} z_1 = \sum_{(i,j)\in E} t^1_{ij} x_{ij} \\ z_2 = \sum_{(i,j)\in E} t^2_{ij} x_{ij} \\ \sum_j x_{aj} - \sum_j x_{ja} = +1 \\ \sum_j x_{gj} - \sum_j x_{jg} = -1 \\ \sum_j x_{ij} - \sum_j x_{ji} = O \quad i \in E\backslash\{a, g\} \end{cases}$$

$$x_{mn} \in \{0, 1\},\ (m, n) \in E$$

Après linéarisation de f, le programme linéaire obtenu obtenu est :

$$\max \sum_{k=1}^{n} w'_k (k r_k - \sum_{i=1}^{n} b_{ik})$$

$$\text{s.c.} \begin{cases} z_1 = \sum_{(i,j)\in E} t^1_{ij} x_{ij} \\ z_2 = \sum_{(i,j)\in E} t^2_{ij} x_{ij} \\ \sum_j x_{aj} - \sum_j x_{ja} = +1 \\ \sum_j x_{gj} - \sum_j x_{jg} = -1 \\ \sum_j x_{ij} - \sum_j x_{ji} = O \quad i \in E\backslash\{a, g\} \\ r_k - b_{ik} \leq -z_i \quad i = 1, 2 \quad k = 1, 2 \end{cases}$$

$$r_k \in \mathbb{R},\ x_{mn} \in \{0, 1\},\ (m, n) \in E,\ b_{ik} \geq 0,\ i = 1, 2,\ k = 1, 2$$

Solution

Après linéarisation de f et résolution du programme linéaire obtenu avec $w = (2, 1)$, nous obtenons $x_{ab} = x_{bc} = x_{cf} = x_{fg} = 1$.

Un chemin robuste pour les deux scénarios est donc (a, b, c, f, g). Le temps correspondant à un tel chemin est de 11 pour le premier scénario et de 8 pour le second. La valeur de la fonction objectif à l'optimum est de -30.

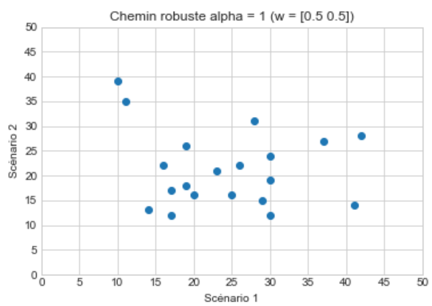
Note : La résolution de ce même programme linéaire avec le vecteur de pondération $w = (1, 0)$ donne $x_{ab} = x_{be} = x_{eg} = 1$. Le chemin robuste pour les deux scénarios correspondant à cete solution est alors (a, b, e, g). Il est de coût 10 pour les deux scénarios. On a ici valorisé une attitude plus égalitariste et légèrement moins utilitariste. La valeur de l'objectif à l'optimum est alors de -10.

4.3 Etude de la pondération w sur la robustesse de la solution

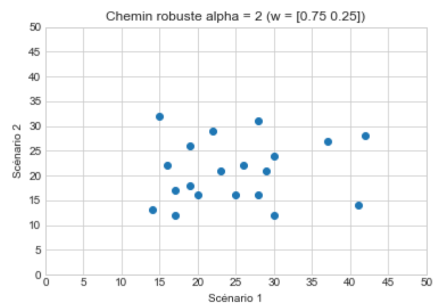
On souhaite désormais étudier l'impact de la pondération w sur la robustesse de la solution proposée.

On utilise une famille de vecteur de pondération $w(\alpha) \in \mathbb{R}^n$ définie pour tout $\alpha \geq 1$ par les poids : $w_i(\alpha) = (\frac{n-i+1}{n})^\alpha - (\frac{n-i}{n})^\alpha$ pour $i = 1, ..., n$.

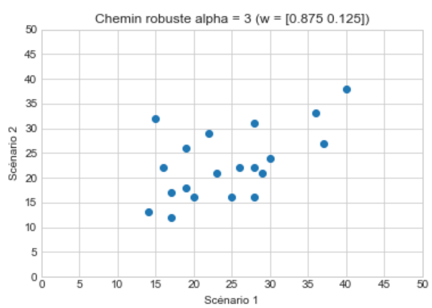
On obtient les résultats suivant pour $n = 2$ et $\alpha = 1, 2, 3, 4, 5$:



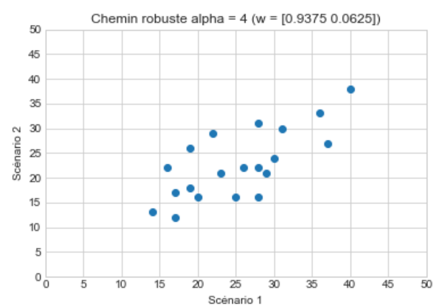
alpha = 1



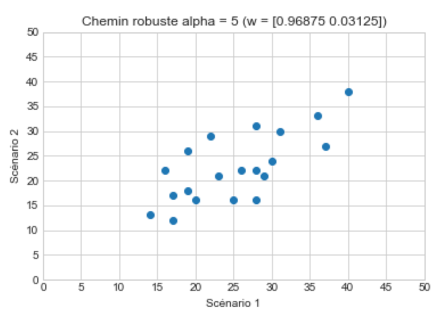
alpha = 2



alpha = 3



alpha = 4



alpha = 5

Figure 4 – Etude de la pondération w sur la robustesse de la solution

On observe que lorsque α augmente, la différence entre la première composante du vecteur poids w et la seconde augmente également.

Graphiquement, cela se traduit par un regroupement des points vers la droite y=x. Ce phénomène traduit le passage d'une attitude très utilitariste lorsque $w = (0.5, 0.5)$ à une attitude de plus en plus égalitariste lorsque $w = (0.9375, 0.0625)$.

En effet, dans le premier cas, certains points montrent clairement qu'un scénario est privilégié par rapport à l'autre (par exemple le temps correspondant au parcours de a à g sera égal à 10 pour le premier scénario alors qu'il sera égal à 40 pour le second). Le but est alors principalement de minimiser le temps moyen ce qui correspond à une attitude utilitariste. Au contraire, lorsque α augmente, les points s'approchent de la droite y=x ce qui traduit le fait que les temps de parcours dans les deux scénarios sont alors relativement similaires ce qui traduit une attitude plus égalitariste.