



ECOLE
POLYTECHNIQUE
DE BRUXELLES

PROJ-H402 - Computing project

PROMETHEE γ GUI

LUKA BONHEURE

2023

Plan du rapport

1	Introduction	3
2	Fonctionnalités	3
2.1	Data tab	3
2.1.1	Possibilités d'améliorations	4
2.2	Result tab	5
2.2.1	Possibilités d'améliorations	7
2.3	HelpForParameters tab	7
2.3.1	Préférence learning	7
2.3.2	Custom module	9
2.3.3	Possibilité d'amélioration	9
2.4	Menu	9
3	Choix des outils	9
3.1	Python	9
3.2	Librairie graphique	9
3.3	Librairies supplémentaires	10
3.4	Documentation	10
3.5	Diagramme UML	10
4	Détails de l'implémentation	10
4.1	Méthode PROMETHEE γ	10
4.2	Preference learning	11
4.2.1	Préférence	11
4.2.2	Indifférence	11
4.2.3	Incomparabilité	12
4.2.4	Algorithme	12
5	Conclusions	14
6	Annexe	15
6.1	Format du fichier csv	15

1 Introduction

Dans le cadre de la recherche sur la relation d'incomparabilité dans la méthode PROMETHEE, une nouvelle version dénommée PROMETHEE γ a été proposée (cf. [DD23]). Cette nouvelle méthode fait intervenir trois nouveaux paramètres T_I , T_J et P_f . L'objectif du projet était de concevoir une application graphique permettant de visualiser l'influence de ces paramètres afin d'aider le décideur à les déterminer. A cette application a été ajoutée la possibilité de déterminer ces paramètres via une procédure d'élicitation (preference learning).

Il est considéré que le lecteur a le code source à disposition (il est en libre accès sur <https://gitlab.com/Luka012/proj-h402-promethee-gamma-gui>) avec une documentation complète. C'est pourquoi ce rapport n'exposera pas de code, mais se concentrera sur l'explication des fonctionnalités et des raisonnements mathématiques.

2 Fonctionnalités

L'application permet de charger des données, de faire fonctionner la méthode PROMETHEE γ sur ces données et de visualiser les résultats de trois manières différentes. Elle propose aussi une aide à la détermination des trois nouveaux paramètres introduits par PROMETHEE γ .

L'application a été conçue pour fonctionner aussi bien sur Windows que sur Linux. Les seules différences observées entre les 2 systèmes d'exploitation sont des modifications mineures au niveau de l'affichage.

L'application se divise en 3 onglets.

2.1 Data tab

Cet onglet (voir Figure 1) permet de charger des données à partir d'un fichier au format csv (détail du format en annexe 6.1) grâce au bouton "Open a File" et d'afficher ces données. Des modifications sur ces données peuvent aussi être effectuées directement à la main.

Only csv files are accepted

Criteria:	g0	g1	g2	g3	
Weights:	0.3333333333333333	0.1666666666666666	0.1666666666666666	0.3333333333333333	Add a criterion Delete a criterion
Preference Functions:	Linear	Linear	Linear	Linear	
Pc:	2.0	2.0	2.0	10000.0	
Qc:	0.5	0.5	0.5	2000.0	
a1	81.3	17.0	14.2	55314.0	
a2	82.8	19.5	12.5	54508.0	
a3	82.3	18.5	12.4	57707.0	
a4	80.9	18.9	12.6	58662.0	
a5	81.9	19.4	12.8	48511.0	
a6	81.6	19.8	12.1	52085.0	
a7	81.5	16.1	12.5	56197.0	
a8	81.3	17.6	12.7	38080.0	

Add an alternative Delete an alternative

Figure 1: "Data tab" avec exemple de données (sur Windows 11, en plein écran)

Les données sont affichées dans une table, reprenant l'ensemble des critères et des alternatives. A chaque colonne correspond un critère contenant les informations suivantes :

- un nom
- un poids
- une fonction de préférence
- p_c et g_c : 2 thresholds, respectivement de préférence et d'indifférence, utilisés par certaines fonctions de préférences.

Le nom de la fonction de préférence est affichée dans un bouton. En cliquant sur ce bouton, une nouvelle fenêtre s'ouvre, affichant les 6 types de fonctions de préférences possible, et permettant de sélectionner celle voulue (Voir Figure 2).

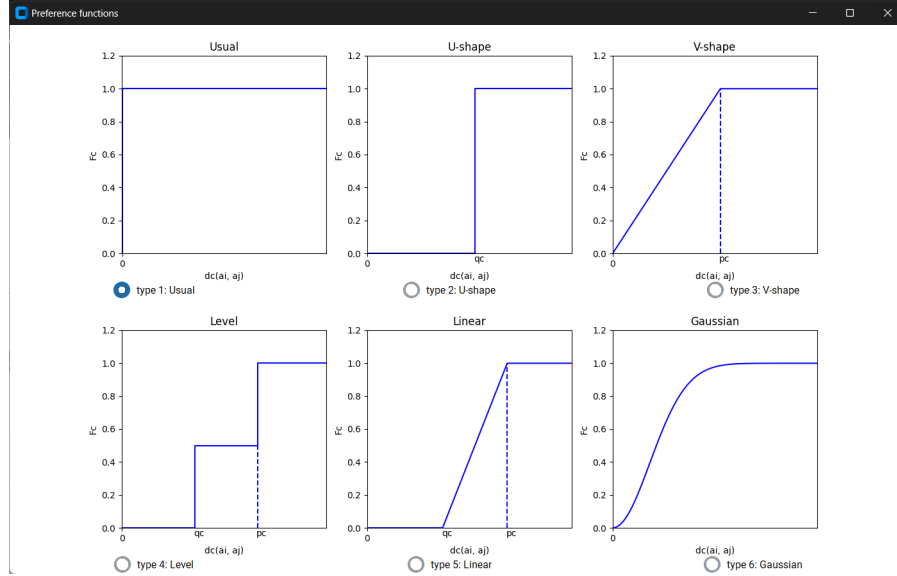


Figure 2: Les 6 types de fonctions de préférence (sur Windows 11)

Chaque alternative est représentée par une ligne, contenant un nom, et l'évaluation de l'alternative pour chaque critère. Des boutons : ajouter une alternative, supprimer une alternative, ajouter un critère, supprimer un critère permettent d'ajouter ou de supprimer des données. L'utilisateur peut naviguer dans la table, quelque soit sa taille, grâce à une scrollbar verticale et une scrollbar horizontale.

2.1.1 Possibilités d'améliorations

- Actuellement, un seul format de fichier est accepté en entrée. Une plus grande diversité de format de fichiers acceptés en entrée augmenterait la flexibilité de l'application et faciliterait le travail des utilisateurs.
- Seul le dernier critère ou dernière alternative peut être supprimé. Une amélioration possible serait de permettre à l'utilisateur de choisir la colonne ou la ligne à supprimer.
- Lors de la navigation avec les scrollbars, l'ensemble de la table est comprise. Une amélioration possible serait de garder les noms des alternatives et des critères constamment visible, afin de toujours pouvoir voir à quelle alternative et pour quel critère correspond une évaluation, même lorsque les données sont nombreuses.

2.2 Result tab

Cet onglet (voir Figure 3) affiche les valeurs des trois nouveaux paramètres introduits par la méthode PROMETHEE γ ainsi que trois manières différentes de visualiser les résultats de la méthode.

Les valeurs des paramètres peuvent être modifiées grâce à des entrées, mais aussi grâce à des curseurs.

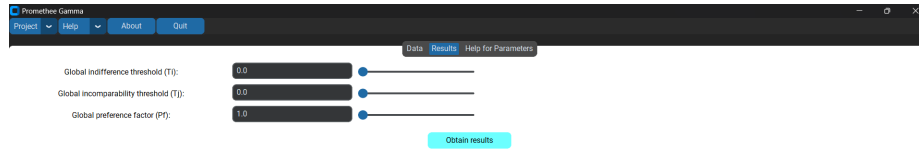


Figure 3: "Results tab" avant l'obtention des résultats (sur Windows 11, en plein écran)

Un bouton "Obtain Results" permet de lancer l'exécution de la méthode et d'afficher les résultats. Le texte du bouton est modifié en "Reload Results" après le premier clic. Il est nécessaire de recharger les résultats si les données ont été modifiées dans l'onglet data, mais pas en cas de modification des valeurs des trois paramètres. En effet, la visualisation des résultats réagit dynamiquement aux changements de valeur des paramètres. Les trois affichages des résultats sont les suivants :

- "Matrix" (voir Figure 4)

Il s'agit de l'affichage le plus simple. La matrice des résultats est une matrice $n \times n$ où n est le nombre d'alternative. Chaque élément ij de la matrice correspond à une relation entre l'alternative i et l'alternative j . Il y a 3 relations différentes possibles : la préférence, l'incomparabilité et l'indifférence. Elles sont notées comme suit : $a_i P a_j$, $a_i J a_j$ et $a_i I a_j$; ce qui signifie : a_i est préféré à a_j , a_i est incomparable à a_j et a_i est indifférent à a_j . Comme la relation de préférence n'est pas symétrique, contrairement à l'incomparabilité et à l'indifférence, une quatrième possibilité existe : $a_j P a_i$, c'est-à-dire, a_j est préféré à a_i .

- "Orthogonal graph" (voir Figure 5)

Ce graphique ne permet pas de visualiser directement les résultats de la méthode PROMETHEE γ , mais un composant de celle-ci : les valeurs de γ associées à chaque couple d'alternatives. La couleur des points indique la relation entre les alternatives correspondantes : rouge pour l'incomparabilité, vert pour l'indifférence et bleu pour la préférence. Ainsi, la proportion de chaque relation et l'impact de chacun des trois paramètres peuvent être rapidement visualisés.

- "Ranking" (voir Figure 6)

Ce graphique permet de visualiser le classement des alternatives. Il est toutefois important de noter que ce n'est pas un classement absolu. La position de chaque alternative, représentée par un cercle contenant son

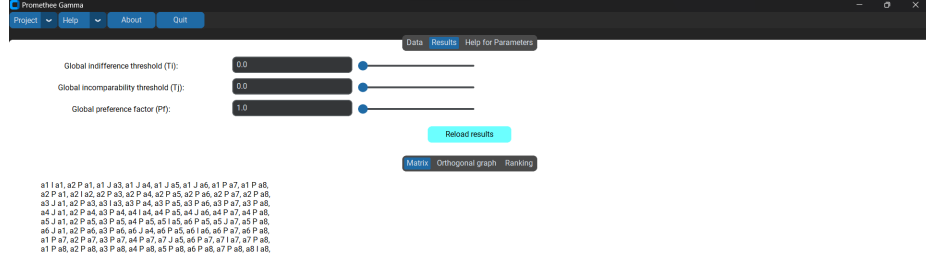


Figure 4: "Results tab" avec affichage de la matrice des résultats (sur Windows 11, en plein écran)

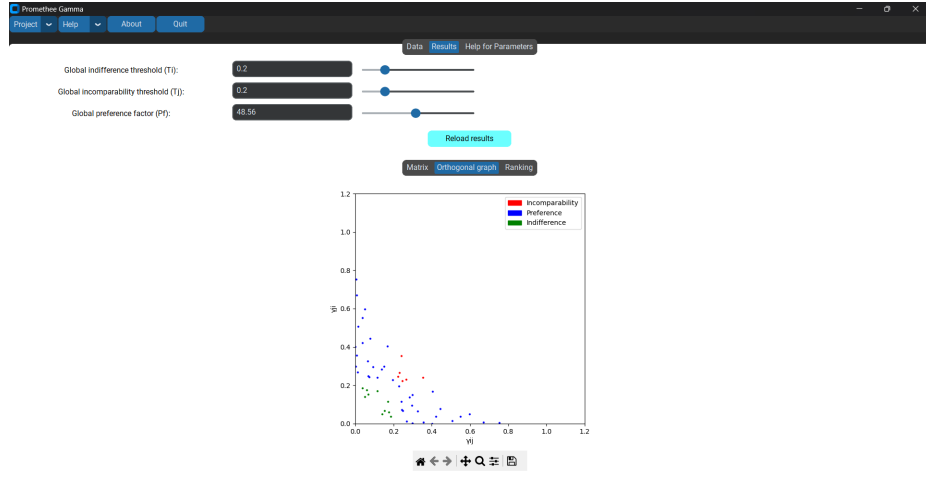


Figure 5: "Results tab" avec affichage du graphe orthogonal (sur Windows 11, en plein écran)

nom, est déterminée par le nombre de fois qu'elle est préférée à une autre alternative. Cela signifie donc que 2 alternatives au même niveau ne sont pas forcément indifférentes ou incomparables. L'une peut être préférée à l'autre. Simplement, elles sont chacune préférées au même nombre d'autres alternatives. De même, des alternatives à des niveaux différents peuvent être indifférentes ou incomparables. Des traits, rouges pour l'incomparabilité, verts pour l'indifférence, relient les alternatives concernées.

Ce graphique permet de visualiser rapidement si une ou plusieurs alternatives en dominent d'autres. Cependant, comme ce n'est pas un classement absolu, l'alternative en tête ne correspond donc pas forcément à la "meilleure" alternative, surtout si elle a des liens d'incomparabilité ou d'indifférence avec d'autres alternatives. Il donne néanmoins une idée générale du "score" que l'alternative a obtenu avec la méthode.

Afin de garder le graphique visible quelque soit sa taille, seule une fraction du graphe est visible à la fois, et des scrollbars permettent de parcourir tout le graphe. Afin d'avoir une vue d'ensemble du graphique, la meilleure solution est d'enregistrer l'image, grâce à la sauvegarde du projet ou avec la barre d'outil se trouvant sous le graphe dans l'application, et de l'ouvrir ensuite avec une autre application d'affichage. Lorsque le nombre d'alternatives croît, le graphe peut aussi perdre en clarté à cause de la superposition

des liens d'indifférence et/ou d'incomparabilité. Ainsi, pour connaître avec certitude la relation entre deux alternatives, la meilleure solution est de se référer à la matrice des résultats. Il est cependant possible aussi de sélectionner certaines alternatives dans la vue à droite du graphe. Les alternatives désélectionnées n'apparaîtront alors plus dans le graphe. Par conséquent, la visibilité augmentera pour les alternatives restantes.

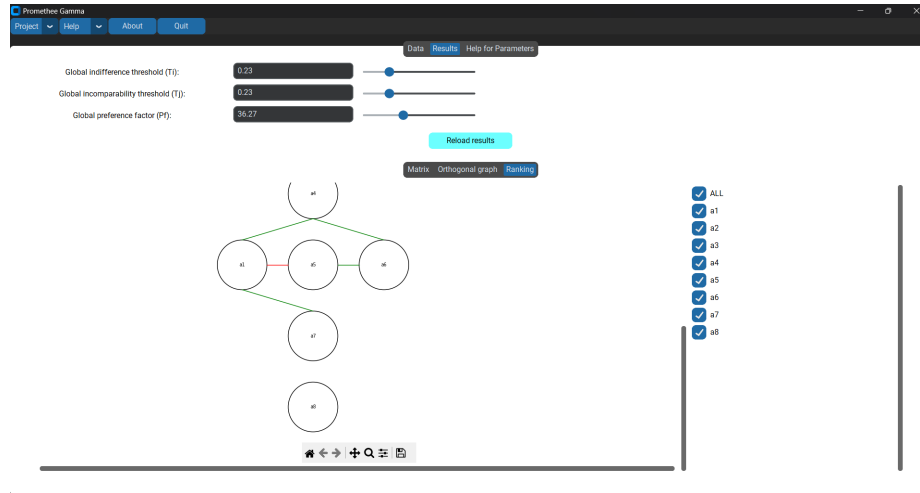


Figure 6: "Results tab" avec affichage du classement (sur Windows 11, en plein écran)

2.2.1 Possibilités d'améliorations

De nombreuses améliorations sont possibles. Par exemple, la matrice des résultats pourrait bénéficier d'un affichage sous forme de tableau, plutôt qu'une simple impression dans une "textbox". Aussi, plutôt que des scrollbars, un système de "dézoom" pourrait être préférable pour le graphe de classement. La possibilité de visionner le graphe dans une fenêtre séparée, afin de pouvoir agrandir l'image sans devoir l'enregistrer et l'ouvrir dans une autre application, pourrait aussi être une bonne solution.

2.3 HelpForParameters tab

Cet onglet (voir Figure 7) propose une aide à la détermination des paramètres introduits par PROMETHEE γ .

L'onglet affiche d'abord 2 boutons, l'un pour entrer dans la méthode de préférence learning, l'autre pour entrer dans le module custom.

2.3.1 Préférence learning

Un algorithme de préférence learning itératif permet de calculer des intervalles de valeurs possibles pour les paramètres. Il utilise les réponses de l'utilisateur pour un petit nombre de comparaisons par paires d'alternatives (10 maximum). Les détails de l'algorithme se trouve dans la section 4.2.

Une fois dans la méthode de préférence learning, 2 nouveaux boutons s'affichent, "Generate questions" et "Quit". "Quit" permet simplement de quitter la méthode et de revenir à l'affichage précédent. "Generate questions" permet de générer la première question (comparaison par paire). Celle-ci s'affiche sous forme de question à choix multiple. L'utilisateur peut choisir une des 4 relations possible (indifférence, incomparabilité, et préférence, dans un sens ou dans l'autre) pour les 2 alternatives affichées. Après avoir cliqué sur "Generate questions", 2 nouveaux boutons s'affichent en-dessous : "Next" et "Confirm". "Next" permet d'afficher la

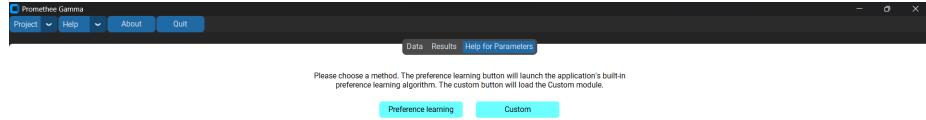


Figure 7: "HelpForParameters tab", affichage de départ (sur Windows 11, en plein écran)

question suivante. Lorsque le nombre maximum de questions est atteint (10 à priori), le bouton "Next" disparaît. "Confirm" clôture l'algorithme. En cliquant sur "Confirm", les boutons "Next" et "Confirm" sont remplacés par des boutons "Use results in results tab" et "Cancel". "Use results in results tab" permet de transférer les résultats de l'algorithme à l'onglet résultats et renvoie sur celui-ci. Ainsi, les résultats de l'algorithme peuvent être directement visualisés dans les résultats de la méthode PROMETHEE γ . Le bouton "Cancel" annule la confirmation et permet de reprendre l'algorithme là où il en était. Chaque comparaison par paire est affichée sous forme d'onglet, et l'utilisateur peut naviguer facilement entre les onglets, et peut ainsi modifier ses réponses. Les résultats de l'algorithme qui s'affichent au-dessus change dynamiquement en fonction des réponses de l'utilisateur.

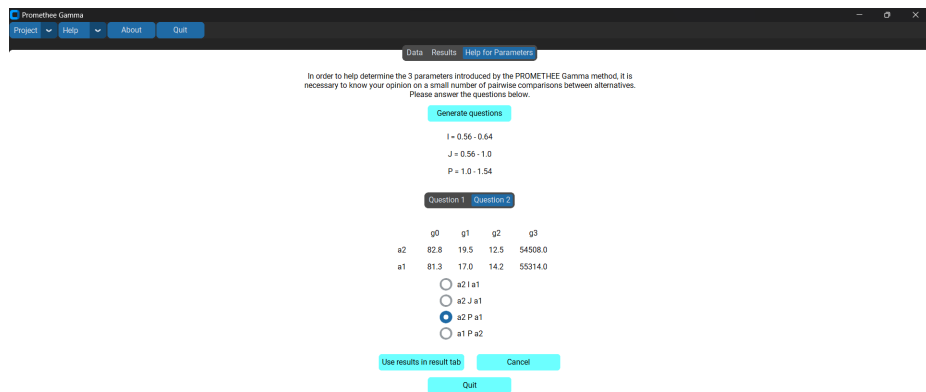


Figure 8: "HelpForParameters tab", preference learning, résultats confirmés après la seconde question (sur Windows 11, en plein écran)

L'algorithme de préférence learning a besoin à la fois des données du "data tab" et des résultats du "results tab". Si l'un de ces éléments manque, un message d'erreur est affiché à l'utilisateur, lui demandant d'effectuer les opérations nécessaire afin de résoudre le problème. De même, si les données et les résultats ne sont pas synchronisés, ce qui se produit lorsque l'utilisateur modifie les données sans rechargé les résultats.

2.3.2 Custom module

Le module custom est un module vide. Une fois dans le module custom, seul un bouton "quit" est affiché pour revenir en arrière. Dans le code source, ce module correspond à un squelette pour pouvoir ajouter facilement une ou plusieurs autres méthodes d'aide à la détermination des paramètres. Le module est conçu de manière à ce qu'aucune modification ne soit nécessaire dans le reste du code source. Cependant, des besoins spécifiques imprévus pourraient le nécessiter malgré tout.

2.3.3 Possibilité d'amélioration

Diverses améliorations peuvent être envisagées, notamment au niveau de l'affichage des comparaisons par paires. Cependant, les principales possibilités d'améliorations viendraient avec l'ajout d'autres méthodes d'aide à la détermination des paramètres. Éventuellement avec une méthode totalement automatique.

Au niveau de l'algorithme de préférence learning, une amélioration intéressante serait de trouver une heuristique pour choisir les comparaisons par paires proposées à l'utilisateur. En effet, actuellement, les comparaisons par pair sont choisies aléatoirement. L'algorithme n'est donc pas entièrement déterministe, ce qui peut poser problème dans certains cadre d'application, les résultats de l'algorithme étant plus difficilement reproductibles.

2.4 Menu

Au dessus des trois onglets principaux se trouve une barre de menus. On y trouve deux menus déroulants, "Project" et "Help", un bouton "About" et un bouton "Quit".

- "Quit" permet de quitter l'application après une demande de confirmation à l'utilisateur.
- "About" affiche une petite présentation de l'application ainsi qu'un lien vers l'article de Gilles DE-JAEGERE et Yves DE SMET, "Promethee γ : a new Promethee based method for partial ranking based on valued coalitions of monocriterion net flow scores." [DD23] qui est à la base de l'application.
- "Project" affiche une série de boutons : "new", "load", "save" et "save as". "new" permet de créer un nouveau projet (en fait, de réinitialiser l'application) ; "load" de charger un projet existant à partir d'un projet précédemment enregistré ; "save" de sauvegarder le projet dans un dossier lié (si aucun dossier n'est encore lié, "save" appelle "save as") ; et "save as" de sauvegarder le projet en choisissant les éléments à sauvegarder, un nom pour le projet et le dossier dans lequel le projet sera sauvegardé. Quatre éléments peuvent être sélectionner pour la sauvegarde : les données, les résultats (les paramètres, la matrice de résultats et la matrices γ) ainsi que les 2 graphiques : l'"Orthogonal graph" et le "Ranking graph".
- "Help" affiche une série de boutons dont les noms correspondent à différentes parties de l'application. Chacun affiche une fenêtre d'explication sur cette partie de l'application.

3 Choix des outils

3.1 Python

Le choix du langage de programmation s'est porté sur python, pour sa simplicité et sa portabilité. Ce langage est aussi l'un des mieux maîtrisé par les parties prenantes.

3.2 Librairie graphique

Afin de réaliser une application graphique, une librairie graphique est nécessaire. tkinter a été choisie pour sa simplicité d'utilisation et pour son intégration. Cette librairie fait en effet partie du package de base de python.

Afin de rendre l'apparence de l'application plus moderne, tkinter est en réalité peu utilisé directement, mais plutôt par l'intermédiaire de customtkinter, développé par Tom Schimanski (voir sur github: <https://github.com/TomSchimansky/CustomTkinter>). Ce fork de tkinter apporte aussi des fonctionnalités plus évoluées, tels que les onglets.

3.3 Librairies supplémentaires

Pour la création des graphiques (orthogonal graph et ranking ainsi que les 6 fonctions de préférence), matplotlib a été utilisé avec numpy. Ces librairies permettent de générer facilement presque n'importe quel graphe. matplotlib vient aussi avec un outil très pratique : une barre à outil permettant de zoomer, enregistrer le graphique sous forme d'image ou utiliser une "main" pour se déplacer dans un graphique.

3.4 Documentation

Afin de générer la documentation, Sphinx a été utilisé. Ce logiciel permet de générer la documentation de manière presque totalement automatisée, et intègre diverses extensions utiles, tels que la visualisation du code source et des références vers d'autres documentations.

3.5 Diagramme UML

Pour générer un diagramme de classe à partir du code source, pyreverse a été utilisé, pour sa simplicité et ses résultats probants.

4 Détails de l'implémentation

4.1 Méthode PROMETHEE γ

La tâche principale de l'application est l'exécution de la méthode PROMETHEE Gamma.

L'implémentation de la méthode utilise les résultats exposés dans l'article de Gilles Dejaegere et Yves De Smet [DD23] page 16 :

$$\phi^c(a_i) = \frac{1}{n-1} \sum_{j=1}^n (\pi_{ij}^c - \pi_{ji}^c)$$

et

$$\gamma_{ij} = \sum_{f^c(a_i) > f^c(a_j)} w_c \cdot (\phi^c(a_i) - \phi^c(a_j))$$

sachant que

$$\pi_{ij}^c = F_c(d_c(a_i, a_j))$$

ce qui représente combien l'alternative a_i est préférée à l'alternative a_j pour le critère c . F_c est la fonction de préférence du critère c , $d_c(a_i, a_j)$ la différence entre les évaluations de a_i et a_j pour le critère c .

Ainsi, pour chaque critère, la matrice des π_{ij}^c est calculée, puis, la liste des $\phi^c(a_i)$. Ceci est réalisé dans la classe "Criterion" (package Models.DataTabModels).

Ensuite, la matrice des γ_{ij} peut être calculée dans la classe "PrometheeGamma" (package Models). Cette même classe s'occupe ensuite de construire les matrices des I_{ij} , P_{ij} et J_{ij} grâce aux formules suivantes :

$$\begin{aligned} I_{ij} &= T_I - \max(\gamma_{ij}, \gamma_{ji}), \\ P_{ij} &= (\gamma_{ij} - \gamma_{ji}) / P_f, \\ P_{ji} &= (\gamma_{ji} - \gamma_{ij}) / P_f, \\ J_{ij} &= \min(\gamma_{ij}, \gamma_{ji}) - T_J. \end{aligned}$$

(page 17 de l'article)

Avant de terminer par la matrice des résultats, déterminée grâce à ces formules-ci :

$$\begin{aligned} a_i I^\gamma a_j &\Leftrightarrow I_{ij} \geq \max(P_{ij}, P_{ji}, I_{ij}, J_{ij}), \\ a_i J^\gamma a_j &\Leftrightarrow J_{ij} \geq \max(P_{ij}, P_{ji}, I_{ij}, J_{ij}), \\ a_i P^\gamma a_j &\Leftrightarrow P_{ij} \geq \max(P_{ij}, P_{ji}, I_{ij}, J_{ij}), \\ a_j P^\gamma a_i &\Leftrightarrow P_{ji} \geq \max(P_{ij}, P_{ji}, I_{ij}, J_{ij}). \end{aligned}$$

(page 18 de l'article)

4.2 Preference learning

L'algorithme de préférence learning est basé sur les deux même résultats que exposés ci-dessus.

L'algorithme effectue le processus inverse. Ainsi,

$$\begin{aligned} T_I &= I_{ij} + \max(\gamma_{ij}, \gamma_{ji}), \\ P_f &= (\gamma_{ij} - \gamma_{ji})/P_{ij}, \\ P_f &= (\gamma_{ji} - \gamma_{ij})/P_{ji}, \\ T_J &= \min(\gamma_{ij}, \gamma_{ji}) - J_{ij}. \end{aligned}$$

Traitons chaque cas séparément

4.2.1 Préférence

Tout d'abord, commençons par comparer une préférence par rapport à l'autre :

$$\begin{aligned} &P_{ij} \geq P_{ji} \\ \Leftrightarrow &(\gamma_{ij} - \gamma_{ji})/P_f \geq (\gamma_{ji} - \gamma_{ij})/P_f \\ \Leftrightarrow &(\gamma_{ij} - \gamma_{ji}) \geq (\gamma_{ji} - \gamma_{ij}) \\ \Leftrightarrow &\gamma_{ij} \geq \gamma_{ji} \end{aligned}$$

On constate que les paramètres T_I , T_J et P_f n'ont pas d'influence sur le choix du sens de la relation de préférence. Ainsi, dans le cadre de cet algorithme, les deux sens de la relation de préférence peuvent être traités comme un seul et unique cas. Il y a préférence lorsque :

$$\begin{aligned} |\gamma_{ij} - \gamma_{ji}|/P_f &\geq T_I - \max(\gamma_{ij}, \gamma_{ji}) \\ |\gamma_{ij} - \gamma_{ji}|/P_f &\geq \min(\gamma_{ij}, \gamma_{ji}) - T_J \end{aligned}$$

Ce qui est équivalent à :

$$\begin{aligned} T_I &\leq |\gamma_{ij} - \gamma_{ji}|/P_f + \max(\gamma_{ij}, \gamma_{ji}) \\ T_J &\geq \min(\gamma_{ij}, \gamma_{ji}) - |\gamma_{ij} - \gamma_{ji}|/P_f \end{aligned}$$

Grâce à ces deux inéquations, on peut définir des limites pour T_I et T_J en fonction de P_f .

4.2.2 Indifférence

$$\begin{aligned} T_i &= I_{ij} + \max(\gamma_{ij}, \gamma_{ji}) \\ I_{ij} &\geq P_{ij} = (\gamma_{ij} - \gamma_{ji})/P_f, \\ I_{ij} &\geq P_{ji} = (\gamma_{ji} - \gamma_{ij})/P_f, \\ I_{ij} &\geq J_{ij} = \min(\gamma_{ij}, \gamma_{ji}) - T_j, \end{aligned}$$

Grâce aux résultats obtenu ci-dessus, on sait qu'on peut fusionner les 2 premières équations. Remplaçons aussi I_{ij} par sa valeur. On obtient :

$$\begin{aligned} T_I - \max(\gamma_{ij}, \gamma_{ji}) &\geq |\gamma_{ij} - \gamma_{ji}|/P_f, \\ T_I - \max(\gamma_{ij}, \gamma_{ji}) &\geq \min(\gamma_{ij}, \gamma_{ji}) - T_J, \\ T_I &\leq T_J \end{aligned}$$

La dernière équation est nécessaire pour faire sens, comme exposés pages 17-18 de l'article.

En réarrangeant l'ordre des termes et en simplifiant $\min(\gamma_{ij}, \gamma_{ji}) + \max(\gamma_{ij}, \gamma_{ji})$ en $\gamma_{ij} + \gamma_{ji}$, on obtient:

$$\begin{aligned} T_I &\geq |\gamma_{ij} - \gamma_{ji}|/P_f + \max(\gamma_{ij}, \gamma_{ji}), \\ T_I + T_J &\geq \gamma_{ij} + \gamma_{ji}, \\ T_I &\leq T_J \end{aligned}$$

La première inéquation permet de différencier le cas d'indifférence par rapport à celui de préférence, la deuxième le cas d'indifférence par rapport à celui d'incomparabilité. La troisième est une contrainte supplémentaire qui doit toujours être respectée.

4.2.3 Incomparabilité

Le développement est fort similaire à celui effectué pour le cas de l'indifférence. C'est pourquoi il ne sera pas exposé. On obtient :

$$\begin{aligned} T_J &\leq \min(\gamma_{ij}, \gamma_{ji}) - |\gamma_{ij} - \gamma_{ji}|/P_f, \\ T_I + T_J &\leq \gamma_{ij} + \gamma_{ji}, \\ T_I &\leq T_J \end{aligned}$$

La première inéquation permet de différencier le cas d'incomparabilité par rapport à celui de préférence, la deuxième le cas d'incomparabilité par rapport à celui d'indifférence. La troisième est une contrainte supplémentaire qui doit toujours être respectée.

4.2.4 Algorithme

Au départ, les intervalles de valeurs possibles sont fixés au maximum suivant le cadre de l'application : $[0,1]$ pour T_I et T_J et $[1, \infty[$ pour P_f . L'algorithme fonctionne de manière itérative. A chaque comparaison effectuée par l'utilisateur, 3 étapes successives sont réalisées avec une quatrième étape possible optionnelle :

1. Créations des intervalles pour les limites de T_I et T_J en fonction de P_f pour la nouvelle paire

Ceci est réalisé grâce aux 2 équations dans le cas de la préférence, ou de la première dans le cas d'indifférence ou d'incomparabilité.

2. Mises à jours des intervalles globaux pour T_I , T_J et P_f

Les valeurs limites de T_I et T_J sont modifiées prioritairement par rapport à celles de P_f . Cependant, ces dernières sont aussi modifiées si nécessaire. Dans le cas où une incompatibilité est détectée, c'est à dire que les mises à jours des intervalles sont impossibles sans établir une autre relation que celle demandée par l'utilisateur pour une comparaison précédente, l'algorithme passe à l'étape 4. Sinon, il passe à l'étape 3.

3. Vérification des contraintes

La contrainte $T_J \geq T_I$ ainsi que la deuxième équation dans le cas d'indifférence ou d'incomparabilité sont vérifiées. Si des modifications des intervalles de valeurs de T_I , T_J et P_f sont nécessaires et possibles, ils sont effectués. Si les intervalles ne doivent pas être modifiés ou que les modifications sont possibles, l'itération est terminée. Dans le cas contraire, une résolution de conflit est nécessaire. On passe alors à l'étape 4.

4. Résolution des conflit

Dans le cas où un conflit survient, c'est-à-dire que deux réponses de l'utilisateur sont incompatibles, il faut trouver la meilleure approximation. Pour ce faire, la méthode de la moyenne au sens des moindres carrés est utilisée.

Un ensemble de valeur de P_f est parcouru, et pour chacune de ces valeurs, les intervalles possibles pour T_I et T_J sont calculés précisément (c'est maintenant possible, puisqu'on dispose d'une valeur unique pour P_f). Ensuite, la moyenne arithmétique entre les extrémités des intervalles pour chaque évaluations de l'utilisateur sont calculées. La moyenne est une approximation remplaçant une intersection entre les intervalles. (Le fait qu'il y ait conflit signifie qu'il n'existe pas d'intersection entre tous les intervalles). Une fitness est alors évaluée pour déterminer la qualité de la solution. Elle est calculée comme suit :

$$fit_I = \sqrt{\sum_k error_{kI}^2}$$

$$\text{Où } \begin{cases} error_{kI} = 0 & \text{si } Avg_I \in int_{T_I k} \\ error_{kI} = dist(Avg_I, int_{T_I k}) & \text{sinon} \end{cases}$$

Avg_I désigne la valeur moyenne pour T_I sur l'ensemble des évaluations, $int_{T_I k}$ l'intervalle de valeur possible pour T_I pour l'évaluation k et $dist(Avg_I, int_{T_I k})$ la distance entre Avg_I et l'extrémité la plus proche de $int_{T_I k}$.

$$fit_J = \sqrt{\sum_k error_{kJ}^2}$$

$$\text{Où } \begin{cases} error_{kJ} = 0 & \text{si } Avg_J \in int_{T_J k} \\ error_{kJ} = dist(Avg_J, int_{T_J k}) & \text{sinon} \end{cases}$$

Avg_J désigne la valeur moyenne pour T_J sur l'ensemble des évaluations, $int_{T_J k}$ l'intervalle de valeur possible pour T_J pour l'évaluation k et $dist(Avg_J, int_{T_J k})$ la distance entre Avg_J et l'extrémité la plus proche de $int_{T_J k}$.

$$fitness = \sqrt{fit_I^2 + fit_J^2}$$

$fitness$ correspond donc à la qualité de la solution. Une fitness plus faible correspond a une meilleure solution.

La valeur de P_f qui a la plus petite valeur de $fitness$ est choisie, et $T_I = Avg_I$ et $T_J = Avg_J$ pour cette valeur de P_f . La contrainte $T_I \geq T_J$ doit toujours être vérifiée.

L'étape 4 clôture l'itération et donne une valeur unique à chacun des trois paramètres. L'algorithme peut alors sembler terminé à l'utilisateur. Cependant, en ajoutant des comparaisons par paires supplémentaires, de nouvelles valeurs viennent jouer dans la moyenne, et ainsi affiner les résultats.

Dans l'idéal, aucun conflit ne doit exister entre les différentes réponses de l'utilisateur. Cependant, il est souvent difficile de définir avec certitude une relation entre deux alternatives (en fait, ce n'est réellement possible que si l'une domine l'autre). Fixer une relation pour une ou deux comparaison paire et puis chercher à éviter les conflits en définissant les autres peut être une bonne manière pour l'utilisateur de trouver les paramètres lui convenants.

5 Conclusions

L'application a été conçue dans le but de rencontrer au mieux les demandes du client, ainsi, toutes les fonctionnalités de bases sont présentes. Cependant, face aux difficultés rencontrées, parfois complexes, et à cause du temps limité, il a fallu faire certaines concessions. C'est la raison pour laquelle il subsiste encore de nombreuses possibilités d'amélioration.

6 Annexe

6.1 Format du fichier csv

Le format est défini comme suit :

- Une ligne commençant par "c" : pour les noms des critères
- Une ligne commençant par "w" : pour le poids des critères
- Une ligne commençant par "f" : pour le type de fonction de préférence des critères
- Une ligne commençant par "p" : pour le threshold de préférence pour la fonction de préférence des critères.
- Une ligne commençant par "q" : pour le threshold d'indifférence pour la fonction de préférence des critères.
- Une ligne par alternative, commençant par son nom, suivi des évaluations pour chaque critères.

L'ordre des lignes n'a pas d'importance. Il est important de noter que les alternatives ne peuvent pas se nommer "c", "w", "f", "p" ou "q", qui sont réservés.

Exemple :

```
c,g0,g1,g2,g3
a1,81.3,17,14.2,55314
a2,82.8,19.5,12.5,54508
a3,82.3,18.5,12.4,57707
a4,80.9,18.9,12.6,58662
a5,81.9,19.4,12.8,48511
a6,81.6,19.8,12.1,52085
a7,81.5,16.1,12.5,56197
a8,81.3,17.6,12.7,38080
w,0.3333333333333333,0.1666666666666666,0.1666666666666666,0.3333333333333333
f,5,5,5,5
p,2,2,2,10000
q,0.5,0.5,0.5,2000
```

References

- [DD23] Gilles Dejaegere and Yves De Smet. “Promethee γ : A new Promethee based method for partial ranking based on valued coalitions of monocriterion net flow scores”. In: *Journal of Multi-Criteria Decision Analysis* 30.3-4 (2023), pp. 147–160. DOI: <https://doi.org/10.1002/mcda.1805>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/mcda.1805>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/mcda.1805>.