



République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: Modèle d'intelligence et de décision (M.I.D)

Thème

Classement de vecteurs de données en utilisant la Dominance Floue

Réalisé par :

- **M. HASSAINE Mohammed Walid**

Présenté le Dimanche 14 Juin 2015 devant le jury composé de MM.

- *M. MERZOUG M. (Président)*
- *Mme. HALFAOUI A. (Encadreur)*
- *M. HADJILA F. (Examineur)*
- *M. SMAHI I. (Examineur)*

Année universitaire: 2014-2015

Dédicace

Je viens tout juste de terminer la rédaction de ce mémoire de fin d'étude, que je m'empresse de le dédier avec un cœur chaleureux, une immense joie et un grand honneur :

*A mes très chers «**PARENTS**» en signe de ma profonde et affectueuse reconnaissance pour leur amour sans mesure, tous les sacrifices, les soutiens, les tolérances et les encouragements qu'ils ont bien voulu me consentir. Tous les mots restent insuffisants pour leur exprimer mes sentiments. J'espère qu'ils accepteront ces quelques lignes en guise de témoignage.*

*A mes sœurs **Hidayet, Amina** que Dieu vous protège et vous garde.*

*A mes nièces **Lylia et Manel**.*

*A ma fiancée **Chafika**.*

*A mes beaux-frères **Hacène et Kamel**.*

*A mes meilleurs amis **Hakim et Sami**.*

A toute ma famille.

A mes amis(e)s et mes fidèles connaissances.

*A toute la promotion **MID**.*

A tous ceux qui m'ont aidé à faire ce travail.

Remerciements

Tout d'abord, je remercie DIEU tout puissant de m'avoir donné la chance de goûter à la saveur du Savoir.

Ce travail de recherche effectué, au sein du Laboratoire des Systèmes et Technologies de l'Information et de la Communication (STIC), de la Faculté des Sciences à l'Université Abou-Bekr Belkaïd–Tlemcen, a été supervisé sous la direction de Monsieur MERZOUG M.

Un grand merci à Madame HALFAOUI A. de m'avoir honoré par son encadrement, pour sa sympathie, son soutien moral et ses connaissances fructueuses qu'elle m'a apporté tout au long de ce travail.

Un merci particulier à notre «Monsieur MERZOUG M.», d'avoir eu l'amabilité d'accepter de présider le jury de cette thèse.

Nos sincères gratitudees à tous les enseignants pour leur efficient accompagnement tout le long de mon cursus universitaire, Pour leur patience et leur compréhension, et particulièrement M. HADJILA F. et M. SMAHI I. d'avoir accepté de faire partie du jury.

J'exprime également mes remerciements les plus sincères à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce mémoire.

Table des matières

Dédicace	2
Remerciements.....	3
Listes des figures.....	7
Listes des tableaux	8
Introduction générale.....	9
Chapitre 1 :.....	12
Concepts et définitions.....	12
1. Introduction.....	13
2. Classement (Ranking)	13
3. Concept de Pareto Dominance	14
4. Skyline.....	14
5. Logique classique et logique floue	14
6. Fuzzification	15
7. Les services web	16
7.1 Définition.....	16
7.2 Qualité des services web QoS	16
7.3 Critères de qualité de service.....	17
7.4 Sélection et classement des services web	18
8 Conclusion	20
Chapitre 2 :.....	21
La fuzzification de pareto Dominance AFDetS et AFDingS	21

1. Introduction.....	22
2. Différence entre "Pareto dominated" et "Pareto dominating"	22
a. Pareto dominated	23
b. Pareto dominating	24
3. Algorithme de Fuzzy Dominated	25
a. Normalisation.....	25
b. Fonction de Fuzzy Dominated (FED).....	26
c. Fonction d'attribution du score au vecteur (AFDS).....	27
d. Fonction de classement	27
e. Ranking.....	28
4. Algorithme de "Fuzzy dominating"	28
a. Fonction de Fuzzy dominating	28
5. Conclusion	29
Chapitre 3 :.....	30
Etudes de cas et expérimentations.....	30
1. Introduction.....	31
2. Cas pratique 1 (Les services web d'envoi de SMS):	31
a. Introduction cas pratique 1	31
b. Application de l'algorithme à l'exemple	33
c. Fuzzification de pareto dominance	33
d. Classement des services	34
e. Présentation de l'application	35
f. Evaluation expérimentale	35

g. Conclusion du cas pratique 1.....	38
3. Cas pratique 2 (Recommandation d'Hôtels):	40
a. Présentation du cas et de l'application.....	40
b. Discussion des résultats avec "Fuzzy dominated".....	43
c. Simulation et étapes de calculs:.....	45
d. Discussion des résultats avec "Fuzzy dominating"	47
e. Différence des résultats	49
4. Conclusion	50
Conclusion générale	51
Bibliographie.....	52

Listes des figures

Figure 1: Exemple de Fuzzification	15
Figure 2: Pareto dominated	23
Figure 3: Pareto dominating.....	24
Figure 4: Courbe de la fonction FUZZY DOMINATED	26
Figure 5: Représentation de l'application JAVA	35
Figure 6: Boutons d'execution de l'algorithme	40
Figure 7 : Représentation de l'Hôtel.....	41
Figure 8: Choix du TOP-K.....	43
Figure 9: Résultats de classement "Fuzzy dominated"	44
Figure 10: Résultats de classement "Fuzzy Dominating"	47
Figure 11: Différents résultats.....	49

Listes des tableaux

Tableau 1: Un ensemble de services web d'envoi de SMS.....	32
Tableau 2: Normalisation des données du Tableau 1	34
Tableau 3: Classement des services selon AFDetS ().....	34
Tableau 4: Top-5 services Rang selon AFDingS, AFDetS () avec $\varepsilon = 0$, $\lambda = 0,2$	36
Tableau 5: Top-5 Services (AFDingS () Vs. AFDetS ()) avec $d = 7$	37
Tableau 6: Top-5 Services (AFDingS () Vs. AFDetS ()) avec $d = 9$	38

Introduction générale

Contexte et problématique

Le classement de vecteurs de données est utilisé dans plusieurs domaines. On veut obtenir un choix des meilleurs hôtels selon plusieurs critères, les meilleurs web services répondant à une requête d'utilisateur selon plusieurs critères de qualité. On doit pouvoir comparer l'ensemble des solutions (vecteur de données candidats) répondant à une requête, les classer et retourner les meilleurs. Cette comparaison peut être faite en utilisant le concept de pareto dominance.

Néanmoins, le concept de pareto dominance ne permet pas toujours d'assurer cette comparaison, en effet, certains vecteurs peuvent être incomparables.

La logique est le premier principe mathématique en informatique, la logique classique permet à l'ordinateur de traiter la donnée et de donner un résultat binaire égal à 0 ou 1.

La logique floue a fait son apparition pour combler le vide laissé par la logique classique permettant ainsi de diversifier les résultats dans un intervalle infini entre $[0,1]$.

La logique floue est, de nos jours, assez utilisée dans plusieurs domaines.

Dans ce projet de fin d'études, nous allons l'utiliser pour la fuzzification du concept Pareto dominance afin de pouvoir classer les vecteurs de données.

L'algorithme de classement nous permettra d'avoir le top-k des vecteurs à travers un score.

Nous pouvons utiliser cet algorithme pour classer des services web, des vecteurs de données, ainsi que dans plusieurs autres domaines.

Contribution :

Notre projet a pour but d'assurer le classement TOP-K des services web et le classement de données réelles en prenant des vecteurs composés de plusieurs critères tel que (la recommandation d'hôtels).

Pour cela, nous proposons une approche constituée de quatre étapes :

- La première effectue une normalisation des données.
- La deuxième consiste en l'utilisation de l'algorithme Fuzzy dominated.
- La troisième affecte un score pour chaque vecteur de données dans sa classe.
- La quatrième réalise un classement ascendant de chaque vecteur.

Plan du mémoire :

Le présent mémoire est composé de trois chapitres et une conclusion générale qui sont organisés comme suit :

Chapitre I :

Ce chapitre est dédié à la présentation et à la définition des concepts de bases utilisés dans notre contexte de travail. Tout d'abord, nous commencerons par donner quelques définitions et présenter la logique floue. Vu qu'un des deux cas pratiques étudié dans ce mémoire sera sur le classement des services web, nous consacrerons une partie à leurs définitions et nous donnerons aussi les travaux actuels qui s'intéressent à leurs classements en utilisant la pareto dominance et la logique floue.

Chapitre II :

Ce chapitre est consacré à la fuzzification de pareto dominance.

Nous présenterons notre algorithme et la différence entre Fuzzy dominated et Fuzzy dominating.

Chapitre III :

Ce chapitre est consacré à l'expérimentation du prototype (Algorithme Fuzzy dominated et Fuzzy dominating) avec deux cas pratiques.

Tout d'abord, nous présentons le cas pratique 1 relatif au classement des services web selon leurs qualités.

Ensuite nous présentons le cas pratique 2 relatif à l'utilisation de données réelles de l'hôtellerie pour la recommandation du meilleur hôtel suivant plusieurs paramètres.

Nous terminerons par une conclusion où l'on résumera les résultats de notre travail et on suggèrera les futures perspectives.

Chapitre 1 :

Concepts et

définitions

1. Introduction

Les concepts et définitions que nous allons examiner dans ce chapitre, nous permettront d'introduire le contexte de notre travail, et nous définirons les concepts clés à savoir les notions de classement de Pareto dominance et de la logique floue qui seront utilisés par la suite dans notre algorithme et applications. La dernière partie de ce chapitre va être consacrée aux web services et aux travaux relatifs à la sélection et au classement des web services car un des cas d'applications de notre travail portera sur le classement des web services basés sur la Qualité de service.

2. Classement (Ranking)

Le classement est une technique qui consiste à ranger dans un ordre méthodique les vecteurs de données selon un ordre préétabli, de façon à pouvoir se procurer facilement et rapidement l'information recherchée.

Principes de classement:

Un bon classement doit être :

Simple, global, évolutif (prend en compte tous les sujets traités),

Applicable à tous les documents,

Valable pour tout le personnel,

Supervisé par 1 seule personne (check-in/check-out).

Il y a plusieurs façons de prévoir et d'organiser un bon classement du fait que les données à classer sont, en général, variées et les personnes, qui recherchent une information, ne portent pas toujours de même données.[1]

3. Concept de Pareto Dominance

La notion de pareto dominance a été définie au 19^{ème} siècle par le mathématicien Vilfredo Pareto elle permet la comparaison entre deux vecteurs de critères. On dit que A domine B si et seulement si A est supérieur ou égale à B pour tous les critères et strictement supérieur pour au moins un critère.

mathématiquement :

Soient deux vecteurs X et Y à n critères, on dit que X domine Y si et seulement si $X_k \geq Y_k$, pour tout $k=1..n$ et $X_k > Y_k$, pour au moins un k.

La relation de dominance Pareto, largement utilisée dans un certain nombre de la prise de décision, peut souffrir d'un manque de pouvoir de discrimination, et qui peut provoquer un grand nombre de solutions non dominées ou optimales.

4. Skyline

Le skyline regroupe l'ensemble des vecteurs de critères (solutions) qui ne sont pas pareto dominée par aucun autre vecteur donc le skyline représente les solutions non dominées ou optimales

5. Logique classique et logique floue

Dans le cadre de la logique classique, une proposition est soit vraie, soit fausse (1 ou 0). Par exemple, la logique classique peut facilement partitionner la température d'une pièce en deux sous-ensembles, «moins de 15 degrés» et «15 degrés ou plus». La figure 1.1a montre le résultat de cette partition. Toutes les températures de moins de 15 degrés sont alors considérées comme appartenant à l'ensemble «moins de 15 degrés». On leur affecte une valeur de 1. Toutes les températures atteignant 15 degrés ou plus ne sont pas considérées comme appartenant à l'ensemble «moins de 15 degrés». On leur attribue une valeur de 0.

Cependant, le raisonnement humain s'appuie fréquemment sur des connaissances ou des données inexactes, incertaines ou imprécises. Une personne placée dans une pièce dont la température est soit de 14.95 degrés soit de 15.05 degrés, ne fera certainement pas de distinction entre ces deux valeurs. Cette personne sera pourtant capable de dire si la pièce est «froide» ou «chaude», sans pour cela utiliser de température limite ni de mesure précise. La logique floue permet de définir des sous-ensembles, comme «froid» ou «chaud», en introduisant la possibilité pour une valeur d'appartenir plus ou moins à chacun de ces sous-ensembles.

La logique floue permet de faire intervenir les notions d'imprécision et d'incertitude dans un système. Cela permet par exemple de faire intervenir une température «d'environ 15 degrés» dans un contrôleur flou. L'incertitude et l'imprécision peuvent également être prises en compte dans le cadre de la logique floue quand on utilise une connaissance issue d'un expert humain. Comment pourrait-on utiliser avec des outils standards une connaissance humaine du genre : «il pleut souvent en hiver» ?[2]

6. Fuzzification

C'est un processus qui consiste à transformer une grandeur numérique en un sous-ensemble flou.

Qualifier une valeur numérique avec un terme linguistique.[3]

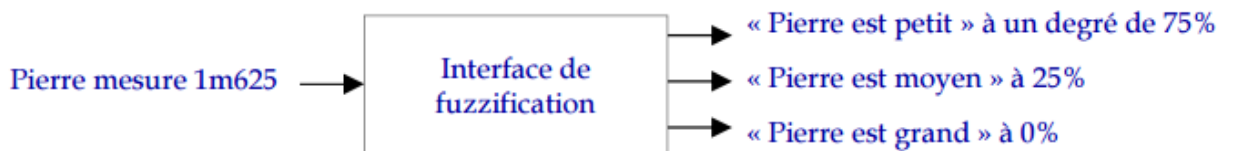


Figure 1: Exemple de Fuzzification

7. Les services web

7.1 Définition

Un service Web est une « unité logique applicative » accessible en utilisant les protocoles standard d'Internet.

Une « librairie » fournit des données et des services à d'autres applications.

Un objet métier qui peut être déployé et combiné sur Internet avec une faible dépendance vis-à-vis des technologies et des protocoles.

Combine les meilleurs aspects du développement à base de composants et du Web.

Un service web présente les caractéristiques suivantes:

- Réutilisable
- Indépendamment de la plate-forme (UNIX, Windows, ...)
- l'implémentation (VB, C#, Java, ...)
- l'architecture sous-jacente (.NET, J2EE, Axis...)[4]

7.2 Qualité des services web QoS

Les services Web sont des systèmes logiciels qui répondent généralement à des besoins fonctionnels. Les contraintes temps réel expriment des exigences temporelles sur le comportement de l'application. Ces exigences sont totalement indépendantes des besoins fonctionnels de l'application. Ainsi les contraintes temps réel seront exprimées à travers des besoins non fonctionnels. Les besoins non fonctionnels des services web s'expriment généralement à travers la qualité de service (QoS).

La QoS est un ensemble de propriétés et de caractéristiques d'une entité ou d'un service qui lui confèrent l'aptitude à satisfaire des besoins déclarés ou implicites.

Ces besoins peuvent être liés à des paramètres tels que l'accessibilité, la disponibilité, le temps de réponse, le coût, la fiabilité, etc. Ces paramètres peuvent être alors considérés comme un critère de choix lorsque l'on doit sélectionner, parmi plusieurs services web découverts ceux qui respectent les contraintes de temps imposées.[5]

7.3 Critères de qualité de service

Dans ce qui suit, nous présentons l'ensemble des attributs de QoS génériques.

Les attributs les plus communs sont décrits par les paramètres liés à la performance, à savoir:

- Le débit : Le nombre de requêtes servies par unité de temps.
- Le temps de réponse : Le temps séparant l'émission de la requête et la réception des résultats.
- La fiabilité : La capacité d'un service d'exécuter correctement ses fonctions.
- La scalabilité : La capacité du service de traiter le plus grand nombre d'opérations ou de transactions pendant une période donnée tout en gardant les mêmes performances.
- La robustesse : La probabilité qu'un service peut réagir proprement à des messages d'entrées invalides, incomplets ou conflictuels.
- La disponibilité : La probabilité d'accessibilité d'un service.
- Le prix de l'exécution : C'est le prix qu'un client du service doit payer pour bénéficier du service.

- La réputation : C'est une mesure de la crédibilité du service. Elle dépend principalement des expériences des utilisateurs finaux.
- La sécurité : C'est un regroupement d'un ensemble de qualités à savoir: la confidentialité, le cryptage des messages et le contrôle d'accès.[6]

7.4 Sélection et classement des services web

Avec la sélection des services web, on cherche à faire le choix du meilleur fournisseur d'un service web, étant donné un ensemble de fournisseurs de ce service. Il y a une proposition de base sur la Qualité de Service (i.e. QoS en anglais Quality of Service – QoS). La plate-forme de cette proposition est un modèle réglé qui peut coexister avec les registres UDDI dérégulés. Les registres dérégulés actuels peuvent fournir des services aux gens pour qui la qualité de service n'est pas importante. Les registres réglés basés sur ce modèle peuvent servir aux applications qui ont besoin de qualité de service.

De nombreux efforts ont été consacrés au problème de la sélection des services Web basée sur QoS. Certains d'entre eux utilisent la technique de programmation linéaire [7], [8]. Les techniques de programmation linéaire sont étendus dans [7] pour inclure des contraintes locales. D'autres travaux utilisent les méthodes combinatoires dans [9] les auteurs utilisent un algorithme heuristique pour résoudre le problème de la sélection des services avec de multiples contraintes de QoS.

Dans [10], les auteurs présentent un algorithme de sélection basé sur une ontologie. Néanmoins, la majorité de ces approches sont plus appropriés pour le nombre limité de services (le processus de sélection a une complexité exponentielle de l'espace) et le nombre limité de QoS, surtout lorsque les utilisateurs doit attribuer des poids sur les attributs QoS.

Dans une recherche récente, le paradigme skyline a été introduit comme un bon mécanisme et efficace pour réduire le nombre de services candidat et de simplifier le processus de sélection. L'idée de skyline provient de

l'ancienne recherche comme le problème de contour, vecteur maximale et la coque convexe et a été introduit dans les bases de données par Börzsönyi [11] qui développe trois algorithmes: BNL, DC et B-tree; ce qui conduit à développer et à améliorer plusieurs autres algorithmes comme SFS [12], de la salsa [13], Zorder, [14] et NN [15]. Certains de ces algorithmes exploitent les structures d'index comme [14], [15] pour améliorer le processus skyline. Toutefois, la taille des solutions appartenant au skyline augmentent sous un nombre élevé de qualité de service et privilège des fois des services parfois avec un mauvais compromis entre la qualité de service.

Pour traiter le problème du skyline, quelques travaux combinent l'avantage de skyline et le ranking et ont défini des variantes de skyline comme [16], [17], [18] et [19]. Dans [17], les auteurs présentent le concept de fréquence skyline qui est le nombre de sous-espaces où un point p est skyline, Dans [16] Chan et al. présentent la notion de k -domination qui étend le Pareto dominance à un sous-ensemble de paramètres k , il existe cependant relation de dominance cyclique (CDR) qui conduit à la perte de skylines en plus k -dominance souvent renvoie un ensemble vide. Dans [19].

Ces approches reposent sur la relation de Pareto dominance ainsi, ils ne privilègent pas des services web avec un bon compromis entre les paramètres, cet inconvénient peut être résolu par la fuzzification de Pareto dominance afin de classer les services incomparables. La fuzzy dominance a été utilisée dans la communauté des bases de données comme [21], les auteurs montrent l'objectif de fuzzification de la notion de dominance Pareto et l'application dans l'optimisation multi-objectif évolutionnaire. D'autres travaux utilisent ce principe et l'appliquent dans l'algorithme génétique ou l'algorithme de particule Swarm. Dans la communauté 'Service computing', [22] utilise la dominance floue et propose l' α -dominance pour classer les services Web basé sur des paramètres de qualité de service et associe le score flou dominant aux services Web. Comme mentionné dans [21] les mesures entre deux vecteurs a, b « a domine b avec un degré α » et « a est dominé par b avec un degré α » n'est pas symétrique, En plus de cela, dans [23], les auteurs

démontrent que l'utilisation de la mesure 'est-dominé' ou 'dominated' est plus efficace dans le classement des services que la mesure 'domine' ou 'dominating'. Notre travail est proche de [22]. Cependant, [22] utilise la relation 'Fuzzy dominating' pour comparer les services au lieu de l'utilisation de la mesure 'Fuzzy dominated' dans le classement des services. A partir de ces observations, nous allons définir dans notre travail la relation Fuzzy dominated 'FDeT et la fonction AFDetS(). Le chapitre suivant présente la définition de ce concept et son utilisation dans notre contexte.

8 Conclusion

Dans ce chapitre nous avons défini les concepts de bases parmi lesquels la notion de pareto dominance sert à comparer un ensemble de vecteurs de données. Nous avons vu aussi que cette notion a été utilisée dans le classement des services web et a été étendu par la logique floue. Nous présentons dans le chapitre suivant la fuzzification de pareto dominance.

Chapitre 2 :

**La fuzzification de
pareto Dominance
AFDetS et AFDingS**

1. Introduction

Dans ce chapitre nous présentons un algorithme de classement qui retourne les meilleurs vecteurs, ces vecteurs peuvent être (services web, données réelles, ou autres).

Cet algorithme repose sur la notion de fuzzification de pareto dominance.

On notera qu'il faudra différencier entre les deux mesures "est pareto dominé" et "pareto domine"; les deux mesures ne sont pas symétriques et ne donnent pas le même résultat.

2. Différence entre "Pareto dominated" et "Pareto dominating"

Prenons un ensemble G , qui contient 3 vecteurs $G=\{A,B,C\}$.

G : Liste de tous les vecteurs.

A, B, C : Vecteurs de données.

Leurs données sont comprises entre 0 et 1 : $[0,1]$.

a. Pareto dominated

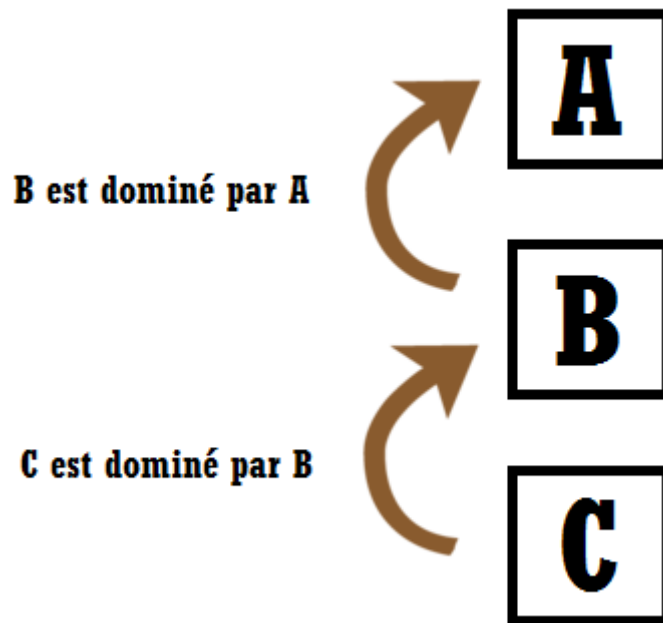


Figure 2: Pareto dominated

Pour ce cas, le classement est : A - B - C.

La chronologie du classement se fait ainsi :

B est dominé par A alors on classe A - B

Puis C est dominé par B alors on classe B - C

au final, on obtient un classement de : A - B - C

b. Pareto dominating

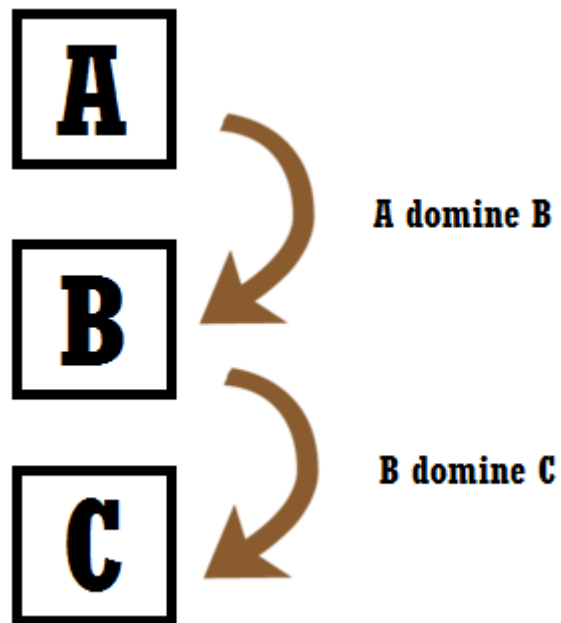


Figure 3: Pareto dominating

Pour ce cas, le classement est : A - B - C.

La chronologie du classement se fait ainsi :

A domine B alors on classe A - B

Puis B domine C alors on classe B - C

au final, on obtient un classement de : A - B - C

3. Algorithme de Fuzzy Dominated

Les différentes étapes de cet algorithme sont:

- Normalisation des vecteurs
- Fed (Fonction fuzzy dominated)
- AFDS (Fonction d'agrégation)
- La fonction de classement
- Ranking

a. Normalisation

La fonction de normalisation consiste à donner un score pour chaque donnée pour un résultat compris entre 0 et 1.

On a l'ensemble G.

On a les données A,B,C,D,E.

Alors $G = \{A,B,C,D,E\}$.

On trouve le Min de G. Supposons que le $\text{Min}(G) = A$.

On trouve le Max de G. Supposons que le $\text{Max}(G) = E$.

$A' = 0$ et $E' = 1$.

Pour trouver le résultat de B,C et D :

$$B' = (B - \text{Min}(G)) / (\text{Max}(G) - \text{Min}(G))$$

le B' est le nouveau chiffre compris entre 0 et 1 qui représente le B.

Exemple:

$G = \{A = 2, B = 4, C = 5, D = 8, E = 10\}$.

$\text{Min}(G) = A = 2$.

$$\text{Max}(G) = E = 10.$$

$$A' = 0 \text{ et } E' = 1.$$

$$B' = (4-2)/(10-2) = 2 / 8 = 0,25.$$

b. Fonction de Fuzzy Dominated (FED)

La fonction de fuzzy dominated est notée FED.

Elle traite deux vecteurs a la fois.

La fonction fuzzy dominated est représentée dans la figure ci-dessous sous la forme suivante :

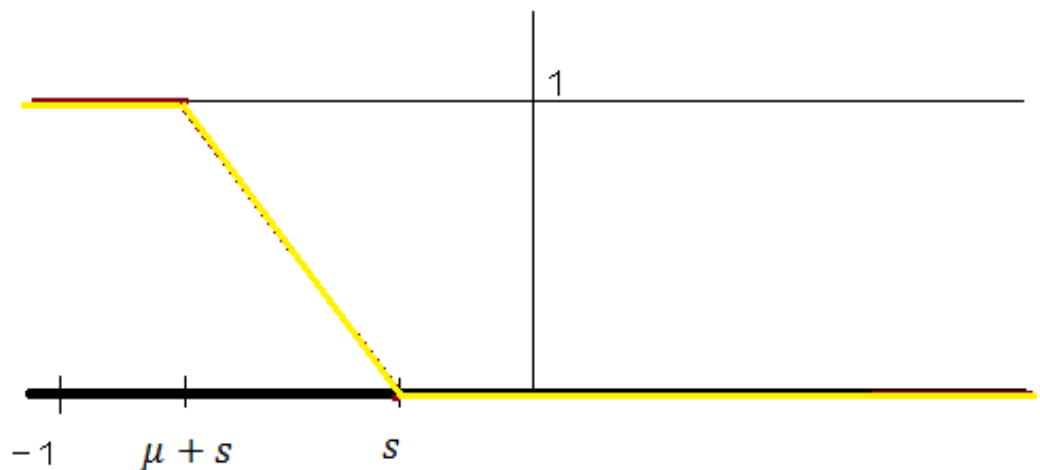


Figure 4: Courbe de la fonction FUZZY DOMINATED

On a deux vecteurs U et V .

$$U = (u_1, u_2, \dots, u_n).$$

$$V = (v_1, v_2, \dots, v_n).$$

n est le nombre de paramètres.

μ et s appartient à $[-1, 0]$ et $(\mu + s) \geq -1$.

$$\mathbf{FED}(U_i, V_i) = \begin{cases} 0 & \text{si } U_i - V_i \geq s \\ \frac{|U_i - V_i - s|}{|\mu + s|} & \text{si } (\mu + s) \leq (U_i - V_i) \leq s \\ 1 & \text{si } U_i - V_i < \mu + s \end{cases}$$

c. Fonction d'attribution du score au vecteur (AFDS)

La fonction d'agrégation est notée AFDS.

Elle agrège les scores élémentaires de la fonction FED.

n est le nombre de paramètres.

$$\mathbf{AFDS}(U, V) = \frac{1}{n} (\sum_{i=1}^n fed(U_i, V_i))$$

d. Fonction de classement

La fonction de classement est notée Rang(U).

Elle permet de classer le vecteur U précédemment appelé.

On a la variable **BASE** qui est le nombre de vecteurs ou de classes ou de services.

$$\mathbf{Rang}(U) = \frac{1}{|\mathbf{BASE} - 1|} (\sum_{(U \neq V)} \mathbf{AFDS}(U, V))$$

e. Ranking

Le ranking permet maintenant de donner le nombre de vecteurs a classé d'ordre croissant du meilleur au moins meilleur, ce nombre est appelé TOP-K. où k représente le nombre des premiers meilleurs vecteurs.

Par exemple:

TOP-5 veut dire qu'on affiche alors les 5 premiers meilleurs vecteurs.

4. Algorithme de "Fuzzy dominating"

Les différentes étapes de cette algorithme sont :

- Normalisation
- Fed (Fonction fuzzy dominating)
- AFDingS (Fonction d'agrégation)
- La fonction de classement
- Ranking

Toutes les étapes de cette algorithme sont identiques à celles de FUZZY DOMINATED. à l'exception de la fonction fuzzy dominating qui est définit comme suit:

a. Fonction de Fuzzy dominating

On a deux vecteurs **U** et **V**.

$$U = (u_1, u_2, \dots, u_n).$$

$$V = (v_1, v_2, \dots, v_n).$$

n est le nombre de paramètres.

μ et s appartient à $[-1,0]$ et $(\mu + s) \geq -1$.

$$\mathbf{FED(U_i, V_i)} = \begin{cases} 0 & \mathbf{si} \quad U_i - V_i \geq \mu \\ 1 & \mathbf{si} \quad U_i - V_i \leq \mu + s \\ & \mathbf{sinon} \quad \frac{|U_i - V_i - \mu|}{s} \end{cases}$$

5. Conclusion

Nous avons présenté dans ce chapitre deux algorithmes de classement Fuzzy dominated et fuzzy dominating. Les étapes et les fonctions relatives à ces deux algorithmes ont été détaillées. Dans le prochain chapitre, nous montrerons l'application de ces deux algorithmes à travers deux cas pratiques grâce à des exemples explicites. Les expérimentations montreront la différence entre les deux concepts à travers les résultats retournés par les deux algorithmes.

Chapitre 3 :

Etudes de cas et expérimentations

1. Introduction

Dans ce chapitre nous allons présenter deux cas pratiques en appliquant les algorithmes définis dans le chapitre précédent.

Le premier concerne le classement des services web basée sur QoS.

Le deuxième concerne la recommandation des hôtels.

Nous présentons dans un premier temps, pour chaque cas, la définition, la conception et la réalisation du système. Nous illustrons les étapes par un exemple illustratif et nous terminerons par une discussion sur les résultats de l'expérimentation.

Les résultats montrent que l'utilisation de l'algorithme fuzzy dominated offre des résultats meilleurs que celui de fuzzy dominating en donnant un avantage aux vecteurs qui ont un bon compromis entre les valeurs de critères.

2. Cas pratique 1 (Les services web d'envoi de SMS):

a. Introduction cas pratique 1

Alors que le Web est rempli avec un nombre considérable de services Web, il existe un grand nombre de fournisseurs de services concurrents qui offrent la même fonctionnalité, mais avec différentes qualités de service (QoS) tels que le temps de réponse, le prix, etc. Par conséquent, QoS est donc un critère essentiel pour choisir parmi les services Web fonctionnellement similaires.

Exemple. Considérons un service Web pour envoyer des SMS, de nombreux services Web fournissent la fonctionnalité (par exemple, ClicSend, Inteltech, etc.), mais avec différents QoS. (Tableau 1) fournit une telle fonctionnalité avec les paramètres de QoS. Ces paramètres de qualités ont été tirés de la base de données ouverte qui contient des données

réelles. La base contient plus de 2500 web services et 9 QoS. Dans notre exemple, chaque service Web a quatre paramètres QoS Q1, Q2, Q3 et Q4, dit respectivement Temps de réponse, débit (ie, Nombre total de invocations / période de temps) Fiabilité (Ratio: nombre de messages d'erreur / messages au total) et les meilleures pratiques (c.-à- , l'étendue d'un WSDL suit une spécification). Pour sélectionner un service Web adéquat, les utilisateurs ont besoin d'examiner tous les services manuellement. L'utilisateur peut également trouver des difficultés à équilibre entre les différentes métriques de qualité. nous avons défini dans le chapitre 1, que le skyline contient l'ensemble des services web qui ne sont pas pareto dominé par les autres. le skyline présente une bonne solution pour réduire le nombre de services Web candidat, et permet de simplifier le processus de sélection du fait qu'il surmonte la limitation majeure des approches actuelles qui nécessitent que l'utilisateur attribue des poids aux différents QoS.

Web service	Provider	Opération	Q1(ms)	Q2(hits/sec)	Q3(%)	Q4(%)
S1	Acrosscommunication.com	SMS	113.8	5.2	81	84
S2	Sjmillerconsultants.com	SMS	179.2	0.7	65	69
S3	Webservicex.net	SendSMS	1308	6.3	67	84
S4	Webservicex.net	SendSMSWorld	3103	5.3	79.3	91
S5	Smsinter.sina.com.cn	SMSWS	751	6.8	64.3	87
S6	Sms.mio.it	SendMessages	291.07	5.2	53.6	84
S7	www.barnaland.is	SMS	436.5	4.5	43.2	84
S8	Emsoap.net	emSoapService	424.54	4.3	11.9	80

Tableau 1: Un ensemble de services web d'envoi de SMS

Selon notre exemple (Tableau 1), nous avons S1 domine S6, S7, S8. Les services S1, S3, S4, S5 appartiennent au skyline et ils ne sont pas comparables entre eux. Nous pouvons remarquer que le skyline réduit le nombre de services candidats, dans notre exemple, nous élimine 50% des services de candidats. Cependant, le nombre de services appartenant au skyline augmente avec l'augmentation de Qos.

Le skyline peut aussi éliminer certains services Web intéressants comme S6 qui est dominé par S4 alors que S4 est le pire service en terme de temps de réponse, cependant S6 a un bon temps de réponse et est plus proche de S4 sur les autres paramètres de qualité de service. Motivé par cela, nous avons proposé une extension de Pareto relation de domination appelé AFDetS Averaged-Fuzzy-Dominated-Score pour associer un score à chaque service et de les classer, selon le chapitre 2.

b. Application de l'algorithme à l'exemple

Dans cette section, nous allons étudier la fuzzification de la relation de Pareto dominance défini dans le chapitre précédent, et montrer son application sur notre exemple (Tableau 1). Pour permettre une mesure uniforme de Web Services, nous d'abord normaliser les différentes valeurs QoS dans l'intervalle [0,1].

c. Fuzzification de pareto dominance

Dans cette partie nous montrons l'application des différentes étapes de l'algorithme dans notre exemple, le tableau 2 donne les résultats de la normalisation.

Web service	Nq1	Nq2	Nq3	Nq4
S1	1	0.74	1	0.68
S2	0.98	0	0.77	0
S3	0.60	0.92	0.80	0.68
S4	0	0.75	0.98	1
S5	0.79	1	0.76	0.82
S6	0.94	0.74	0.60	0.68
S7	0.89	0.62	0.45	0.68
S8	0.90	0.59	0	0.50

Tableau 2: Normalisation des données du Tableau 1

d. Classement des services

Dans cette partie, nous montrons l'application des différentes étapes de l'algorithme sur notre exemple, le tableau 3 donne les résultats du classement des services selon AFDetS.

Web service	AFDetS	Nq1	Nq2	Nq3	Nq4
S1	0,071	1	0.74	1	0.68
S5	0,107	0.79	1	0.76	0.82
S6	0,143	0.94	0.74	0.60	0.68
S3	0,25	0.60	0.92	0.80	0.68
S7	0,286	0.89	0.62	0.45	0.68
S4	0,312	0	0.75	0.98	1
S8	0,393	0.90	0.59	0	0.50
S2	0,571	0.98	0	0.77	0

Tableau 3: Classement des services selon AFDetS ()

Nous pouvons observer que le service S1 est est top-1. En effet il est mieux que les autres dans Q1, Q2 et a une bonne valeur dans les autres paramètres de qualité de service.

e. Présentation de l'application

Dans l'application JAVA que nous avons réalisée avec NetBeans, sert a retourner l'ensemble des meilleurs services web en les classant selon leurs AFDetS(), on a des entrées et des sorties, la figure suivante présente l'écran principale.

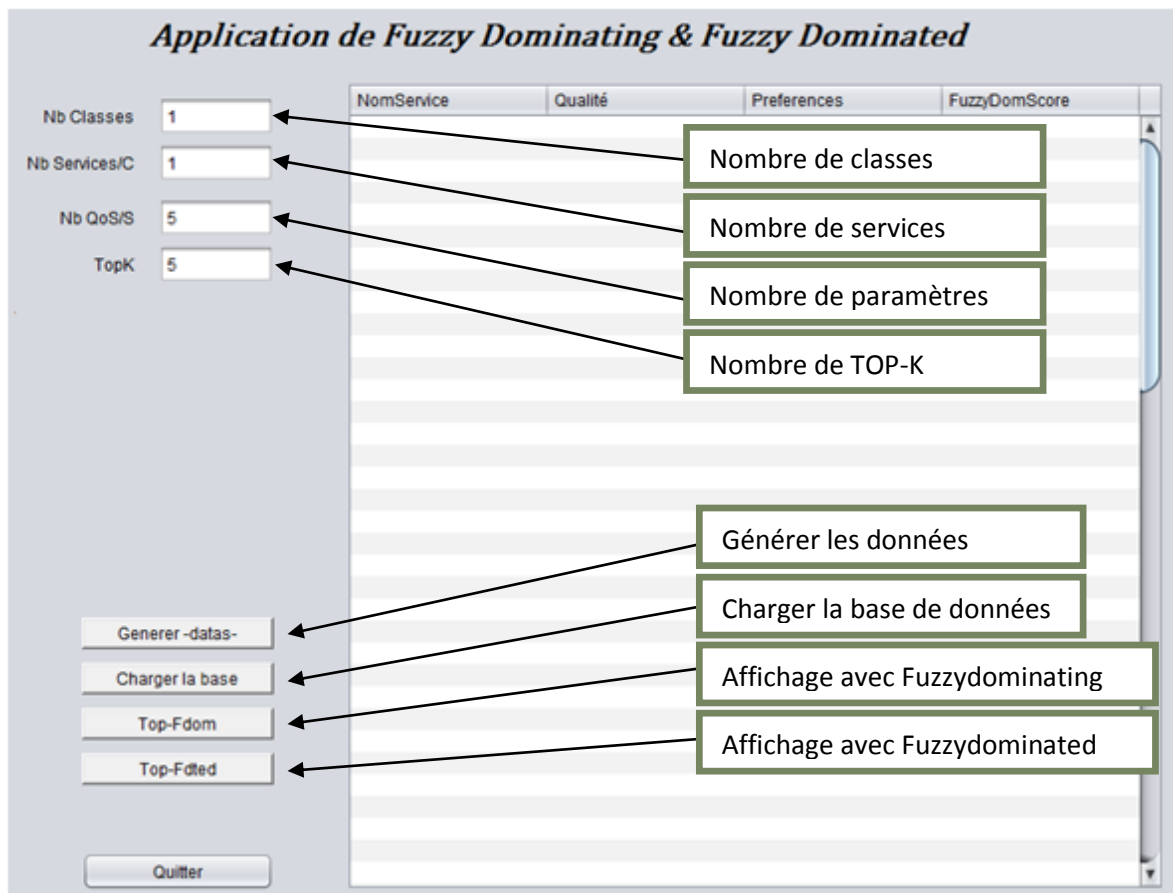


Figure 5: Représentation de l'application JAVA

f. Evaluation expérimentale

Afin d'évaluer et de prouver l'efficacité de notre approche, nous comparons le résultat de l'utilisation de Fuzzy-Dominted avec le score Fuzzy-dominating. La fonction fuzzy-dominating a été aussi utilisée dans les

travaux de [22] on l'appelle dans notre application AFDingS et la comparons à notre algorithme AFDetS.

Plusieurs simulations ont été réalisées en faisant varier les paramètres suivants:

- $E, \lambda, -d$: nombre de paramètres QoS, - n : nombre de services de la même classe S.

Pour chaque simulation, nous prenons les services TOP-5 générées par les algorithmes AFDetS et AFDingS et nous les comparons.

Nous avons utilisé la base de données réelle et publique qui comprend environ 2507 informations de services Web dans le monde réel. Chaque service comprend la mesure des 9 paramètres de QoS. Le nom du service et son adresse WSDL sont également incluses dans l'ensemble de données.

Nous groupons les services fonctionnellement similaires en classes, par exemple le groupe "sms" (envoi de SMS) contient 30 services réels. "Rechercher" (ie. Les services Web moteur de recherche tels que Google Search, Amazone, etc.) contient 92 services.

- **variation de ϵ et λ** : Nous présentons ci-dessous deux scénarios (Tableau 4) en variant ϵ et λ sur un ensemble de 30 services appartenant à la classe de SMS. Chaque service dispose de 4 paramètres de QoS.

<i>Top-5 FDingS</i>			<i>Top-5 FDetS</i>		
<i>ServiceID</i>	<i>FDingS</i>	<i>Qos(q1,q2,q3,q4)</i>	<i>ServiceID</i>	<i>FDetS</i>	<i>Qos(q1,q2,q3,q4)</i>
5	0,566	[0.787,1,0.758,0.818]	12	0,071	[1,0.738,1,0.682]
4	0,551	[0,0.754,0.975,1]	5	0,107	[0.787,1,0.758,0.818]
12	0,529	[1,0.738,1,0.682]	6	0,143	[0.941,0.738,0.603,0.682]
30	0,423	[0.6,0.918,0.797,0.682]	30	0,25	[0.6,0.918,0.797,0.682]
6	0,329	[0.941,0.738,0.603,0.682]	7	0,286	[0,0.754,0.975,1]

Tableau 4: Top-5 services Rang selon AFDingS, AFDetS () avec $\epsilon = 0, \lambda = -0,2$

Nous pouvons observer à partir des résultats (tableau 4) que le classement donné par AFDetS est plus intéressant que celui donné par AFDingS même si nous varions ϵ et λ , le top-1 est toujours meilleur selon AFDetS.

Nous pouvons dire que AFDetS favorise les services avec un bon compromis entre les valeurs de qualité dans tous les paramètres. AFDetS place en bas du classement les services qui ont des valeurs nulles ou qui tendent vers le nul dans certains paramètres même s'ils ont des valeurs élevées dans d'autres paramètres.

- **variant d et n**: Nous présentons ci-dessous deux scénarios en faisant varier d du 7 au 9 sur un ensemble de 92 services appartenant à la classe 'recherche'. Nous fixons $\epsilon = -0,1$ et $\lambda = -0,2$. Le résultat des Top-5 services fournis par l'algorithme AFDingS et AFDetS sont présentés dans (Tableau 5) et (Tableau 6).

	<i>ServiceID</i>	<i>Score</i>	<i>Qos(q1,q2,q3,q4,q5,q6,q7)</i>
Top-5 (AFDingS)	70	0,409	[0.183,0.904,0.618,0.964,0.767,1,0.815]
	30	0,388	[0.164,0.904,1,0.964,0.767,1,0.815]
	24	0,385	[0.005,1,0.829,1,0.767,1,0.667]
	72	0,381	[0.474,0.795,0.260,0.807,0.767,0.667,0.815]
	16	0,365	[0.003,1,0.419,1,1,0.667,0.111]
Top-5 (AFDetS)	30	0,005	[0.164,0.904,1,0.964,0.767,1,0.815]
	52	0,006	[0.016,0.819,0.955,0.94,0.767,1,0.667]
	24	0,006	[0.005,1,0.829,1,0.767,1,0.667]
	70	0,008	[0.183,0.904,0.618,0.964,0.767,1,0.815]
	45	0,022	[0.042,0.831,0.382,0.94,0.767,1,0.667]

Tableau 5: Top-5 Services (AFDingS () Vs. AFDetS ()) avec d = 7

<i>ServiceID</i>	<i>Score</i>	<i>Qos(q1,q2,q3,q4,q5,q6,q7,q8,q9)</i>
------------------	--------------	--

Top-5 (AFDingS)	24	0,397	[0.05,1,0.829,1,0.767,1,0.667,0.004,0.958]
	16	0,366	[0.003,1,0.419,1,1,0.667,0.111,0.030,0.358]
	60	0,344	[0.016,0.988,0.955,1,0.333,1,0.259,0.008,0.337]
	55	0,328	[0.179,0.916,0.244,0.976,0.767,1,0.815,0.066,0.800]
	70	0,318	[0.183,0.904,0.618,0.964,0.767,1,0.815,0,0.021]
Top-5 (AFDetS)	24	0,006	[0.05,1,0.829,1,0.767,1,0.667,0.004,0.958]
	45	0,018	[0.042,0.831,0.382,0.940,0.767,1,0.667,0.030,0.937]
	55	0,018	[0.179,0.916,0.244,0.976,0.767,1,0.815,0.066,0.800]
	52	0,024	[0.016,0.819,0.955,0.940,0.767,1,0.667,0.017,0.105]
	30	0,027	[0.064,0.904,1,0.964,0.767,1,0.815,0.092,0.053]

Tableau 6: Top-5 Services (AFDingS () Vs. AFDetS ()) avec $d = 9$

Sur le (Tableau 5), nous pouvons observer que le classement donné par AFDetS est plus intéressant que celui donné par AFDingS. Le top-1 (AFDetS) est le service 30. Ce dernier a une meilleure valeur que les top-1 (AFDingS) sur q3. En outre, le service 30 est proche du service 7 sur le paramètre de Q1. Nous pouvons remarquer que le service 6 est inclus dans le top-5 (AFDingS) alors qu'il ne fait pas partie des top-5 (AFDetS) en raison de ses mauvaises valeurs sur Q3 et Q7. En fait, il est remplacé par un service 45 qui a un bon compromis entre les paramètres de qualité de service. Voyons maintenant le classement avec $d = 9$ (Tableau 5). Les deux méthodes de classement ont le même top-1 (Service 24). Cependant, les autres services donnés par AFDetS sont différents de ceux fournis par AFDingS. Les services 16 et 70 qui appartiennent au (top-5 (AFDingS)) sont rejetés par AFDetS du top-5 parce qu'ils contiennent certaines mauvaises valeurs (proche / ou égal à 0) sur certains critères QoS. Ces deux services sont respectivement remplacés par le service 45 et le service 30 par l'approche AFDetS, nous pouvons remarquer que ces deux services présentent un bon compromis entre leurs paramètres de qualité de service.

g. Conclusion du cas pratique 1

Dans ce cas pratique, nous avons présenté une application pour le classement des services web basé sur QoS. Nous avons appliqué une fuzzification de la Pareto-domination et appliqué le concept AFDetS qui associe un score à un service selon la relation fuzzy dominated. Nous avons démontré que le concept fuzzy dominated peut offrir une alternative pour comparer les services quand ils sont non comparables avec Pareto dominance. Les résultats expérimentaux montrent que l'approche proposée est prometteuse en comparaison avec le classement donnée par Fuzzy dominating.

3. Cas pratique 2 (Recommandation d'Hôtels):

a. Présentation du cas et de l'application

Un utilisateur voulant choisir un hôtel en entrant certains paramètres aura l'opportunité d'avoir une recommandation suivant ses besoins et les meilleurs résultats avec le site conçu.

Nous avons construit un site qui permet de saisir une requête et qui retourne un classement suivant les deux algorithmes présentés dans le chapitre précédent.

Nous avons différents boutons et chaque bouton affiche un résultat.



Figure 6: Boutons d'exécution de l'algorithme

Les trois boutons affichés sur la figure 6 permettent différents utilités :

- Classement avec fuzzy dominated:
Exécute l'algorithme fuzzydominated et affiche un tableau suivant le nombre Top-k.
- Classement avec fuzzy dominating:
Exécute l'algorithme fuzzy dominating et affiche un tableau suivant le nombre Top-k.
- Voir les deux résultats:
Exécute les deux algorithmes fuzzy dominated et fuzzy dominating et affiche deux tableaux suivant le nombre Top-k.

Pour ce cas de recommandation d'hôtels, nous avons plusieurs données sous forme de vecteurs non-normalisés.

Ce cas nous permettra de mieux comprendre le fonctionnement de l'algorithme de fuzzy dominated.

Dans l'application web, nous trouverons les représentations suivantes :



Figure 7 : Représentation de l'Hôtel

Pour cette Figure 7 : Représentation de l'Hôtel, il existe des paramètres tel que (Paramètre d'avis, d'étoile, Distance, Prix).

Et d'autres données de présentation et de nomination tel que (Titre, photo, validation de la note, et la jauge de vote).

- Paramètre d'avis :

est défini par la moyenne des avis pour cette Hôtel, variant entre 0 et 5.

Plus la moyenne des avis augmente plus l'importance augmente.

Donc on aura un avis moyen compris entre 0 et 5.

Exemple :

On a 3 avis, 1er avis (3/5), 2ème avis (2/5) et le 3ème avis (5/5).

La moyenne des avis est : $(3.33/5)$.

On l'as obtenu avec $((3/5)+(2/5)+(5/5))/3 = 3.33/5$.

- Paramètre d'étoile:

Ce paramètre désigne la qualité de l'Hôtel suivant certains paramètres internationaux compris entre 0 et 5.

Plus le nombre d'étoile augmente plus l'importance augmente.

- Paramètre de distance du centre ville:

Ce paramètre désigne une distance du centre ville plus la distance augmente plus l'importance diminue.

Exemple:

On a trois distances:

1ère distance = 1km. 2ème distance = 5km. 3ème distance = 3km.

La meilleur distance et la première qui est égale a 1km.

Normalisons le problème:

1ère = 1. 2ème = 0.

3ème = $(3-1)/(5-1) = 2/4 = 0.5$.

- Paramètre de prix d'une nuitée:

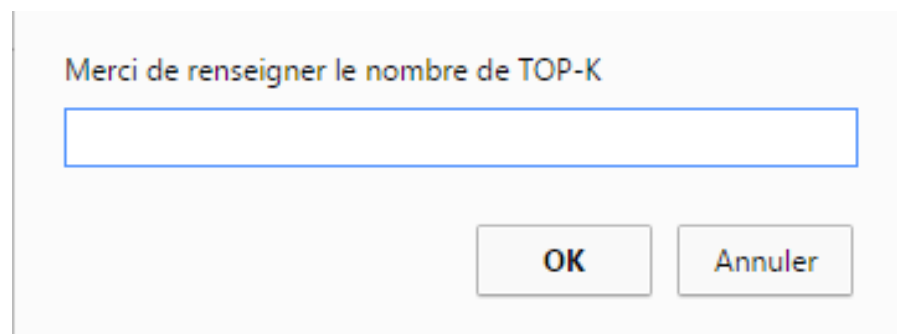
Ce paramètre désigne un prix d'une nuitée, plus le prix augmente plus l'importance diminue. même chose pour la distance.

On conclut que on a 2 paramètres ascendants et 2 paramètres descendants.

b. Discussion des résultats avec "Fuzzy dominated"

Lors du clique sur le bouton classement avec fuzzy dominated.

On demande à renseigner le nombre de TOP-K.



Merci de renseigner le nombre de TOP-K

OK Annuler

Figure 8: Choix du TOP-K

Pour notre cas on affiche les TOP-10

Notre résultat est le suivant:

Classement des top-k avec l'algorithme de "fuzzy dominated"

Rang	Nom de l` Hôtel	Score	Poids du prix	Poids de l` etoile	Poids de la distance	Poids des avis
1	ZIANIDES	0.058823529411765	0.93478	0.75	0.93333	0.44298
2	Renaissance Tlemcen	0.088235294117647	0.56522	1	0.73333	0.75877
3	SHERATON D'ORAN	0.11764705882353	0.52174	1	0.86667	0.60526
4	HAKIM	0.11764705882353	0.47826	1	0.86667	0.65789
5	IBIS Tlemcen	0.14705882352941	0.88261	0.25	1	0.65789
6	Aurassi Alger	0.16176470588235	0.26087	1	0.8	0.74561
7	STAMBOULI	0.20588235294118	0.82609	0.25	1	0.36404
8	AGADIR	0.22058823529412	0.97826	0.5	1	0.32895
9	STAMBOULI 2	0.23529411764706	0.76087	0.5	0.93333	0.2807
10	IBIS ORAN	0.25	0.74348	0.25	0.73333	0.26316

Figure 9: Résultats du classement avec "Fuzzy dominated"

Nous pourrions interpréter le classement de "Fuzzy dominated".

Pour le **score**, plus il est petit plus il est meilleur.

Par exemple on prend:

Top(1) : ZIANIDES.

Score = 0.0588.

Le vecteur des poids de Top(1) est (0.934 , 0.75 , 0.933 , 0.442).

et,

Top(2) : Renaissance Tlemcen.

Score = 0.0882.

Le vecteur des poids de Top(2) est (0.565 , 1 , 0.733 , 0.758).

et

Top(3) : SHERATON D'ORAN.

Score = 0.1176.

Le vecteur des poids de Top(3) est (0.521 , 1 , 0.866 , 0.605).

On constate que les 3 premiers rangs aucun de leurs paramètres ne tend vers le 0.

c. Simulation et étapes de calculs:

On prend 3 vecteurs U,V,T:

$U = (0.875, 0.75, 0.933, 0.4)$

$V = (0.5, 0.95, 1, 0.635)$

$T = (0.083, 1, 0.866, 0.811)$

$S = -0.2; M = 0$.

Etape 1: Calcul du FED(U1,V1)

$U1 = 0.875$, $V1 = 0.5$

$U1 - V1 = 0.375$

$FED(U1, V1) = 0$ car $U1 - V1 \geq -0.2$

Alors on calcul FED(U2,V2), FED(U3,V3), FED(U4,V4).

$FED(U2, V2) = 0$ car $-0.2 \leq (U2 - V2) \leq -0.2$;

$FED(U3, V3) = 0$;

$FED(U4, V4) = 1$.

On calcul aussi

$FED(U1, T1) = 0$,

$FED(U2, T2) = 1$,

$FED(U3, T3) = 0$,

Chapitre 3 : Etudes de cas et expérimentations

$$FED(U_4, T_4) = 1,$$

Etape 2: Calcul AFDS(U,V)

$$AFDS(U,V) = 1/4 * (0+0+0+1).$$

$$AFDS(U,V) = 0.25.$$

On calcul aussi AFDS(U,T) = 0.5.

Etape 3: Calcul du RANG(U)

$$Rang(U) = (1/(3-1)) * (AFDS(U,V)+AFDS(U,T)).$$

$$Rang(U) = 0.375.$$

On calcul aussi

$$RANG(V) = 0.125.$$

et RANG(T) = 0.25.

Etape 4: Classement

Le classement des trois vecteurs est :

V , T , U car classement ascendant de 0 a 1.

$$V = 0.125 , T = 0.25 , U = 0.375.$$

d. Discussion des résultats avec "Fuzzy dominating"

Même chose pour "Fuzzy dominating" on choisi le TOP-K à 10.

Les mêmes étapes de fuzzy dominated se répèterons.

On aura les résultats suivants :

Classement des top-k avec l'algorithmme de "fuzzy dominating"

Rang	Nom de l` Hôtel	Score	Poids du prix	Poids de l` etoile	Poids de la distance	Poids des avis
1	ZIANIDES	0.61764705882353	0.93478	0.75	0.93333	0.44298
2	AGADIR	0.60294117647059	0.97826	0.5	1	0.32895
3	IBIS Tlemcen	0.55882352941176	0.88261	0.25	1	0.65789
4	Renaissance Tlemcen	0.54411764705882	0.56522	1	0.73333	0.75877
5	SHERATON D'ORAN	0.5	0.52174	1	0.86667	0.60526
6	HAKIM	0.5	0.47826	1	0.86667	0.65789
7	STAMBOULI	0.48529411764706	0.82609	0.25	1	0.36404
8	SHERATON D'ALGER	0.48529411764706	0.52609	1	0	1
9	Aurassi Alger	0.48529411764706	0.26087	1	0.8	0.74561
10	BOUAKEL	0.47058823529412	1	0	1	0.13596

Figure 10: Résultats du classement avec "Fuzzy Dominating"

Nous pourrons interpréter le classement de "Fuzzy dominating".

Pour le **score**, plus il est grand plus il est meilleur.

Par exemple on prend:

Top(1) : ZIANIDES.

Score = 0.6176.

Le vecteur des poids de Top(1)est (0.934 , 0.75 , 0.933 , 0.442).

et,

Top(2) : AGADIR.

Score = 0.6029.

Le vecteur des poids de Top(2) est (0.978 , 0.5 , 1 , 0.328).

et

Top(3): IBIS Tlemcen.

Score = 0.5588.

Le vecteur des poids de Top(3) est (0.882 , 0.25 , 1 , 0.657).

et

Top(5) : SHERATON D'ORAN.

Score = 0.4772.

Le vecteur des poids de Top(5) est (0.083 , 1 , 0.866 , 0.811).

On remarque que SHERATON D'ORAN a de meilleurs paramètres par rapport à IBIS Tlemcen car IBIS Tlemcen a un paramètre qui tend vers 0. et pour le classement, il n'est pas logique d'avoir un TOP-K qui a un paramètre qui tend vers 0.

Alors on peut conclure que AFDting favorise des vecteurs avec des paramètres qui tendent vers 0.

Contrairement à Fuzzy Dominated, il n'existe pas de vecteurs TOP-10 qui contiennent des paramètres qui tendent vers 0 et il favorise dans le haut de son classement les vecteurs avec un bon compromis entre les critères.

e. Différence des résultats

On prend TOP-k = 10.

On aura les résultats suivants:

Classement des top-k avec l'algorithme de "fuzzy dominating"

Rang	Nom de l' Hôtel	Score	Poids du prix	Poids de l' étoile	Poids de la distance	Poids des avis
1	ZIANIDES	0.61764705882353	0.93478	0.75	0.93333	0.44298
2	AGADIR	0.60294117647059	0.97826	0.5	1	0.32895
3	IBIS Tlemcen	0.55882352941176	0.88261	0.25	1	0.65789
4	Renaissance Tlemcen	0.54411764705882	0.56522	1	0.73333	0.75877
5	SHERATON D'ORAN	0.5	0.52174	1	0.86667	0.60526
6	HAKIM	0.5	0.47826	1	0.86667	0.65789
7	STAMBOULI	0.48529411764706	0.82609	0.25	1	0.36404
8	SHERATON D'ALGER	0.48529411764706	0.52609	1	0	1
9	Aurassi Alger	0.48529411764706	0.26087	1	0.8	0.74561
10	BOUAKEL	0.47058823529412	1	0	1	0.13596

Classement des top-k avec l'algorithme de "fuzzy dominated"

Rang	Nom de l' Hôtel	Score	Poids du prix	Poids de l' étoile	Poids de la distance	Poids des avis
1	ZIANIDES	0.058823529411765	0.93478	0.75	0.93333	0.44298
2	Renaissance Tlemcen	0.088235294117647	0.56522	1	0.73333	0.75877
3	SHERATON D'ORAN	0.11764705882353	0.52174	1	0.86667	0.60526
4	HAKIM	0.11764705882353	0.47826	1	0.86667	0.65789
5	IBIS Tlemcen	0.14705882352941	0.88261	0.25	1	0.65789
6	Aurassi Alger	0.16176470588235	0.26087	1	0.8	0.74561
7	STAMBOULI	0.20588235294118	0.82609	0.25	1	0.36404
8	AGADIR	0.22058823529412	0.97826	0.5	1	0.32895
9	STAMBOULI 2	0.23529411764706	0.76087	0.5	0.93333	0.2807
10	IBIS ORAN	0.25	0.74348	0.25	0.73333	0.26316

Figure 11: Différents résultats

Interprétation de la différence des résultats.

Pour le Fuzzy dominating, nous constatons que le Top(2) : AGADIR n'existe pas dans la liste des Top-5 de Fuzzy dominated.

et on prend aussi pour Fuzzy dominated que le Top(4) : HAKIM n'existe pas dans la liste des Top-5 de Fuzzy dominating.

Alors nous comprenons entre le Top(2) de fuzzy dominated et le Top(2) de Fuzzy dominating qui ne sont pas les mêmes vecteurs.

$$AFDting(2) = (0.9782, 0.5, 1, 0.3289)$$

$$AFDted(2) = (0.5652, 1, 0.733, 0.7587)$$

pour le AFDted (2) le paramètre minimal = 0.5652 contrairement a AFDting(2) il existe un paramètre = 0.3289.

Nous pouvons conclure la même chose que dans le 1er cas pratique:

Que l'algorithme de Fuzzy Dominated n'accepte pas de Top-K qui ont un paramètre qui tend vers 0.

Si on prend un autre exemple aussi AFDting (3) et AFDted (3).

On trouve une différence sur un paramètre pour AFDting (3) = 0.25 et pour AFDted (3) = 0.521.

Malgré que pour certains cas, nous trouvons des vecteurs de données classés parmi les Top-10 mais cela provient du problème que la BDD n'est pas assez riche.

4. Conclusion

Comme dans les résultats du premier cas d'étude, on confirme que l'utilisation de Fuzzy dominated avantage les vecteurs de données avec un bon compromis entre les paramètres et offre un meilleur classement que celui donné par la fuzzy dominating.

Conclusion générale

Nous avons présenté dans ce travail une approche pour le classement des vecteurs de critères basé sur la fuzzification de Pareto dominance. Cette dernière permet de classer les vecteurs incomparables avec Pareto dominance et favorise dans le haut du classement les vecteurs qui ont un bon compromis entre leurs valeurs.

Nous avons proposé deux algorithmes : FUZZY DOMINATED et FUZZY DOMINATING avec un exposé démonstratif des différences entre les deux concepts.

Nous avons présenté deux exemples d'applications de cette approche, le premier est relatif au domaine de classement des services web et le deuxième permet le classement d'hôtels.

Les résultats des données de l'expérimentation effectuée avec les deux applications réalisées ont montré que la fuzzification de Pareto permet d'attribuer un score au vecteur et d'étendre le degré avec lequel un vecteur de critères domine un autre.

Nous avons démontré que les deux concepts 'fuzzy dominated' et 'fuzzy dominating' ne sont pas symétriques et que le premier offre un classement meilleur que le deuxième.

Les concepts présentés peuvent être utilisés dans des algorithmes d'optimisation multicritères dans la phase locale de sélection donc on prévoit de les tester dans la composition des services web.

Bibliographie

- [1] Mme Aicha ZAHRI, "Le classement," vol. 1, 2008.
- [2] Mr. DJAABOUB Salim, "Logique floue et SMA," no. 354/ Mag /2009, 2009.
- [3] Sabeur ELKOSANTINI, "Introduction à la logique floue," 2010.
- [4] Jeremy Fierstone, "Les services Web," pp. 4-5, Novembre 2002.
- [5] "T. Kossentini, L.Baccouche et H. H Ben Ghezala, Etude de Problème de".
- [6] GHAF FOUR Ayoub, "SELECTION DES SERVICES WEB," UABT, p. 27, 2013.
- [7] " L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, Qos-aware middleware for web services composition, IEEE Trans. Software Eng., vol. 30, no. 5, pp. 311327, 2004. ".
- [8] " D. Ardagna and B. Pernici, Adaptive service composition in flexible processes, IEEE Trans. Software Eng., vol. 33, no. 6, pp. 369384, 2007. ".
- [9] "T. Yu, Y. Zhang, and K.-J. Lin, Efficient algorithms for web services selection with end-to-end qos constraints, TWEB, vol. 1, no. 1, 2007. ".
- [10] " X. Wang, T. Vitvar, M. Kerrigan, and I. Toma, A qos-aware selection model for semantic web services, in ICSOC, 2006, pp. 390401. ".

- [11] Borzsonyi, S., D. Kossmann, and K. Stocker, The Skyline Operator, in Proceedings of the 17th International Conference on Data Engineering 2001, IEEE Computer Society. p. 421-430..
- [12] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, Skyline with Presorting, Proc. 19th IEEE Intl Conf. Data Eng. (ICDE), pp. 717- 816, 2003..
- [13] I. Bartolini, P. Ciaccia, and M. Patella, Efficient Sort-Based Skyline Evaluation, ACM Trans. Database Systems, vol. 33, no. 4, pp. 1-45, 2008..
- [14] K. C. K. Lee, B. Zheng, H. Li, and W.-C. Lee, Approaching the skyline in z order, in VLDB, 2007, pp. 279-290..
- [15] Kossmann, D., F. Ramsak, and S. Rost, Shooting stars in the sky: an online algorithm for skyline queries, in Proceedings of the 28th international conference on Very Large Data Bases 2002, VLDB Endowment: Hong Kong, China. p. 275-286.
- [16] Chan, C.Y., et al., On high dimensional skylines. Advances in Database Technology- EDBT 2006, 2006: p. 478-495.
- [17] Chan, C. Y., et al., Finding k-dominant skylines in high dimensional space. 2006: p. 503-514..
- [18] Vlachou, A. and M. Vazirgiannis, Ranking the sky: Discovering the importance of skyline points through subspace dominance relationships. Data and Knowledge Engineering, 2010. 69(9): p. 943-964.

- [19] Lin, X., et al., Selecting stars: The k most representative skyline operator. 2007: p. 86-95..
- [20] M. Alrifai, D. Skoutas, and T. Risse, Selecting skyline services for qos-based web service composition, in WWW, 2010, pp. 1120..
- [21] M. Koppen and R. Vicente Garcia. A fuzzy scheme for the ranking of multivariate data and its application. In Proceedings of the 2004 Annual Meeting of the NAFIPS (CD-ROM), pages 140145, Banff, Alberta, Canada, 2004. NAFIPS.
- [22] D. Benslimane, and A. Hadjali. A fuzzy framework for selecting top-k Web services compositions. Applied Computing Review, 2011. K. Benouaret,.
- [23] D. Skoutas, D. Sacharidis, A. Simitsis, and T. Sellis. Ranking and clustering web services using multi-criteria dominance relationships. In IEEE Trans. on Services Computing , 2010..
- [24] Zekri Lougmiri, "Résolution efficace des requêtes Top-k par agrégation," vol. I, 2002.
- [25] " E. Al-Masri and Q. H. Mahmoud, Investigating web services on the world wide web, in WWW, 2008, pp. 795804."
- [26] "K. Benouaret, D. Benslimane, and A. HadjAli, On the use of fuzzy dominance for computing service skyline based on qos, in ICWS, 2011, pp. 540547 ".

Résumé

Le classement Top-K des vecteurs de critères est utilisé dans plusieurs domaines comme celui de sélection des services web. En effet, les utilisateurs ont toujours besoin de meilleurs services web qui répondent à leurs besoins, et qui offrent un compromis entre les différents paramètres recherchés.

Le concept pareto dominance et celui du skyline permettent de comparer et de réduire le nombre de vecteurs (solution) retournés aux utilisateurs, néanmoins certains vecteurs restent incomparables et d'autres vecteurs intéressants sont éliminés du classement.

Pour résoudre cette problématique, nous avons proposé la fuzzification de pareto dominance et avons présenté l'algorithme fuzzy dominated.

Nous avons démontré l'efficacité de notre approche à travers deux applications, l'une dédiée au classement des services web basée sur QoS et l'autre réservée au classement des hôtels.

Mots clés:

La logique floue, Fuzzy dominated, Fuzzy dominating, classement, Top-k, skyline.

Abstract

Top-K ranking data vectors is used in several areas such as selection of web services. Indeed, users always need the best web services that meet their needs, and offer a compromise between different parameters sought.

The concept of Pareto dominance and the skyline are used to compare and reduce the number of vectors returned to users, however, certain vectors are incomparable and other interesting vectors are eliminated from the ranking.

To solve this problem, we proposed the fuzzification Pareto dominance and introduced the fuzzy Dominated algorithm.

We have demonstrated the effectiveness of our approach through two applications, one dedicated to the classification of web services-based QoS and the other reserved for hotels ranked.

Keywords:

Fuzzy logic, Fuzzy dominated, Fuzzy dominating, ranking, Top-k, skyline.

ملخص

يستخدم أعلى K معايير تقديم ناقلات في عدة مجالات مثل اختيار الخدمات على شبكة الإنترنت. في الواقع، مستخدمين دائما في حاجة إلى أفضل خدمات الإنترنت التي تلبي احتياجاتهم، وتقديم حل وسط بين المعايير المختلفة المطلوبة. يتم استخدام مفهوم باريتو الهيمنة وأفق للمقارنة وتقليل عدد ناقلات (الحل) عاد للمستخدمين، إلا أن بعض ناقلات هي يتم استبعاد ناقلات لا تضاهي وأخرى مثيرة للاهتمام من الترتيب. لحل هذه المشكلة، اقترحنا fuzzification باريتو الهيمنة وقدم خوارزمية يهيمن غامض. لقد أثبتنا فعالية نهجنا من خلال اثنين من التطبيقات، وجودة الخدمة المستندة إلى الخدمات احدة مخصصة لتصنيف الويب والآخر مخصص للفنادق في المرتبة. الكلمات المفتاحية: المنطق الضبابي، غامض يسيطر عليها، تهيمن ضبابي، لتحتمل المرتبة، أعلى-K، أفق.