Une approche basée sur les flots pour optimiser la sélection de l'alignement des ontologies

Chahira Touati , Moussa Benaissa , Yahia Lebbah

Université d'Oran 1 Ahmed Ben Bella, B.P. 1524 El-M'Naouar, 31000 Oran, Algérie
chahira40@yahoo.fr, moussabenaissa@yahoo.fr et ylebbah@yahoo.fr

RÉSUMÉ. *Les ontologies ont été créées pour résoudre le problème de l'hétérogénéité des données sur le web et pour partager des connaissances de domaine entre les systèmes. Cependant plusieurs ontologies de même domaine sont développées sur le web et sont devenues elles-mêmes source d'hétérogénéité. L'alignement des ontologies est une solution pour résoudre ce type de problème. Il a pour but la découverte des correspondances sémantiques entre des ontologies. Nous présentons dans ce papier une approche basée sur les graphes pour aborder le problème d'alignement des ontologies. Notre approche consiste à modéliser le problème de l'extraction d'un alignement qui satisfait des contraintes de cardinalités comme un problème de minimisation de coût sur un réseau de flots. Pour évaluer notre approche, nous avons utilisé deux types de données (données synthétiques et données réelles) et nous avons comparé notre approche avec les deux algorithmes les plus largement utilisés pour résoudre le problème de l'alignement d'ontologies (l'algorithme de Karp et l'algorithme Hongrois).*

ABSTRACT. *Ontologies have been created to solve the problem of the heterogeneity of data on the Web and to share domain knowledge between systems. However, several ontologies of the same domain are developed on the Web which became themselves source of heterogeneity. The Ontology alignment is a solution to solve this type of problem. It aims to discover the semantic correspondences between ontologies. We present in this paper an efficient graph-based approach to tackle the problem of extracting ontology alignment. More precisely, our approach consists in modeling the problem of extracting an alignment (matching) which satisfies multiple cardinality constraints, as minimizing some cost on a flow network. Our approach has been evaluated on a variety of synthetic and real data, and compared with current used algorithms (e.g., Hungarian and Karp algorithms).*

MOTS-CLÉS : *ontologies, alignement d'ontologies, approche basée sur les graphes, contraintes de cardinalités, réseau de flot.*

KEYWORDS: *ontologies, ontology alignment, graph based approach, cardinality constraints, flow network.*

DOI:10.3166/RIA.30.733-758

1. Introduction

Nées des besoins de représentation des connaissances, les ontologies sont à l'heure actuelle au cœur des travaux menés dans le web sémantique. Elles visent à établir des représentations à travers lesquelles les machines peuvent manipuler la sémantique des informations. Les ontologies sont considérées comme une solution au problème de l'hétérogénéité des données sur le web. Cependant, les ontologies disponibles peuvent elles-mêmes présenter des hétérogénéités : étant données deux ontologies, la même entité peut être donnée sous des noms différents ou simplement être définie de différentes manières (Euzenat et Valtchev (2004))(voir la figure1). Aborder ce problème nécessite d'identifier les correspondances sémantiques entre les entités de différentes ontologies. Ce processus est défini comme l'alignement d'ontologies (Euzenat et Shvaiko (2013)).

Étant données deux ontologies, l'alignement produit un ensemble de correspondances chacune liant deux entités (par exemple, des concepts, des instances, des propriétés, des termes, etc.) par une relation sémantique (équivalence, subsomption, incompatibilité, etc.), éventuellement munie d'un degré de confiance. L'ensemble des correspondances, aussi appelé alignement, peut par la suite être utilisé pour fusionner les ontologies, migrer des données entre ontologies ou traduire des requêtes formulées en fonction d'une ontologie vers une autre (Kengue *et al*. (2008)).

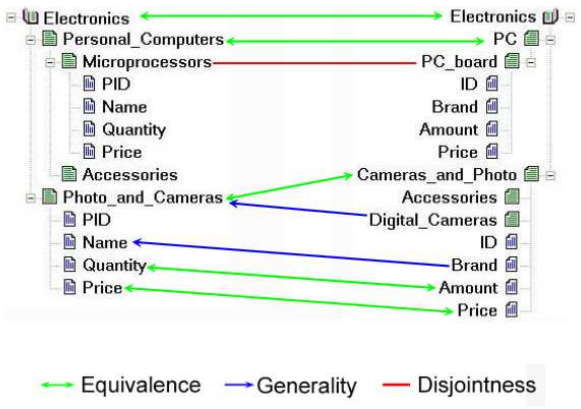


Figure 1. Alignement entre deux ontologies (Shvaiko et Euzenat (2005b))

L'alignement d'ontologies est une opération critique dans de nombreuses applications, y compris l'ingénierie ontologique, l'intégration de l'information, le partage de l'information dans un système pair à pair, la navigation et la recherche d'information sur le web (Euzenat et Shvaiko (2013)).

Diverses solutions au problème d'alignement ont été proposées jusqu'ici, voir (Shvaiko et Euzenat (2005a); Noy (2004); Kalfoglou et Schorlemmer (2003); Euzenat et Shvaiko (2013)), tandis que quelques exemples d'approches individuelles portant

sur le problème de la correspondance peuvent être trouvés dans (Giunchiglia *et al*. (2007); Zhang et Bodenreider (2007)).

Les critères qui sont largement utilisés pour évaluer les systèmes d'alignement sont la précision, le rappel et la f-mesure. Ces critères ont été optimisés dans de nombreux systèmes (Euzenat et Shvaiko (2013); Xingsi *et al*. (2015)). Avec l'apparition du web sémantique et la complexité de plus en plus croissante des ontologies (leurs nombres et volumes), une nouvelle mesure à savoir la performance en termes de temps de calcul est actuellement considérée (Jiménez-Ruiz et Grau (2011)). Le travail présenté dans cet article s'inscrit dans ce cadre et propose d'optimiser l'extraction de l'alignement en termes de temps d'exécution, tout en préservant sa qualité.

L'idée de ce papier consiste à exploiter la théorie des graphes et plus particulièrement la théorie des flots pour résoudre efficacement le problème de l'extraction d'un alignement qui satisfait des contraintes de cardinalités et qui a la plus grande valeur de similarité globale. Il faut noter à ce niveau que le travail présenté ici aborde le problème de l'extraction d'un alignement qui possède des propriétés formelles. Par contre, Il n'aborde pas l'aspect de calcul des similarités qui sont supposées calculées par ailleurs.

Ce papier est organisé comme suit : premièrement nous commençons par donner quelques travaux connexes. Dans la section 3, nous présentons quelques notions préliminaires sur l'alignement des ontologies nécessaires pour comprendre la section suivante. Nous présentons successivement les notions de correspondance, alignement, propriétés d'alignement, mesures de similarité. Dans la section 4, nous décrivons les différents algorithmes utilisés dans notre travail. Dans la section 5, nous détaillons notre approche qui consiste à proposer un modèle basé sur la théorie des flots pour résoudre le problème de l'extraction automatique d'un alignement qui satisfait des contraintes de cardinalités et ayant une similarité globale maximale. Nous commençons cette section par la description des étapes de construction du réseau sur lequel il faut appliquer l'algorithme de flot à coût minimum pour sélectionner l'alignement avec les propriétés requises. Ensuite nous détaillons notre plateforme informatique. Finalement nous présentons nos résultats expérimentaux et nous concluons notre papier.

2. Travaux connexes

Beaucoup de travaux de recherche dans l'alignement d'ontologies sont liés à la question du calcul et de définition des mesures de similarité, alors que peu d'attention a été accordée à la question de savoir comment extraire efficacement l'alignement final d'une matrice de valeurs de similarité. Dans cette section, nous allons examiner les systèmes connexes avec un accent particulier sur les méthodes d'extraction. Globalement on peut distinguer deux approches pour traiter le problème de l'extraction de l'alignement.

1. Approche interactive : l'utilisateur est impliqué dans le processus d'extraction de l'alignement. Une façon de mettre en oeuvre cette approche consiste à afficher

toutes les paires d'entités avec leurs mesures de confiance et celles qui sont jugées les plus pertinents par l'utilisateur sont sélectionnés. Cette approche semble plus pertinente que celle automatique en particulier dans les applications traditionnelles où de grands ensembles de données sont traitées (Shvaiko et Euzenat (2008)). Dans ce cas, nous pouvons citer (Do et Rahm (2007); Noy et Musen (2002)). Les méthodes de cette catégorie sont particulièrement utiles dans les situations où l'identification d'alignements de haute précision est requise. Toutefois, elles présentent l'inconvénient majeur d'être très fastidieuses pour l'utilisateur et sujettes à beaucoup d'erreurs. Cette situation est due essentiellement à la complexité sans cesse croissante des ontologies sur le web (leurs volumes et leurs nombre). Plus précisément, la difficulté réside notamment dans la compréhension des ontologies objets de l'alignement par l'utilisateur d'un côté et du nombre importants des correspondances sémantiques à analyser par l'expert d'un autre côté. Cette approche néanmoins regagne de l'intérêt et représente actuellement l'une des pistes privilégiées pour la recherche dans le domaine de l'alignement des ontologies (Bellahsene *et al*. (2011)). Dans le cas de cette approche, nous pouvons aussi citer des systèmes des éditions récentes de OAEI 2015, comme le travail de (Jiménez-Ruiz *et al*. (2012)). ce travail a montré que la simulation des interactions de l'utilisateur avec un taux d'erreur de 30 % au cours du processus d'alignement a conduit aux mêmes résultats que l'alignement interactif.

2. Approche automatique : les techniques de cette catégorie permettent de surmonter les difficultés inhérentes à l'approche interactive et réduisent la charge de l'utilisateur quant à l'identification et la maintenance de l'alignement entre ontologies. Quelques approches existent dans la littérature pour opérer l'extraction automatique de l'alignement. La méthode la plus simple consiste à filtrer les correspondances candidates (hypothèses) selon un seuil de similarité donné (Euzenat et Shvaiko (2013)). Toutefois, cette technique présente l'inconvénient majeur de ne pas prendre en considération les dépendances entre les hypothèses (Meilicke et Stuckenschmidt (2015)). Le travail décrit dans (Jean-Mary *et al*. (2009)) est du même type et consiste à vérifier que l'alignement final ne contient pas des schémas indésirables prédéfinis.

Le problème d'alignement de l'ontologie a déjà été abordé en exploitant des techniques d'apprentissage automatique. Nous citons les travaux suivants :

- Dans (David *et al*. (2007)) un apprentissage multi-stratégie a été utilisé pour obtenir des instances similaires des hiérarchies pour extraire des concepts similaires utilisant la classification naïve bayésienne.

- Dans (Bagher *et al*. (2006)), à la suite d'un processus d'optimisation des paramètres sur les classifieurs SVM, et réseaux de neurones (NN), un alignement initial a été effectué. Ensuite, les commentaires de l'utilisateur ont été exploités pour améliorer la performance globale.

- L'idée de (Eckert *et al*. (2009)) est tiré de (David *et al*. (2007)) avec un ensemble de données presque similaire. Cette étude calcule 10 mesures de similarités par des méthodes basées sur les chaînes, linguistiques et à base d'instances. Les classifieurs DT et naïve bayésienne ont été appliqués pour classer les échantillons d'entrée.

D'autres approches modélisent le problème de la sélection de l'alignement final comme un problème d'optimisation où un sous-ensemble de correspondances sémantiques candidates, qui maximise une certaine fonction objective, doit être sélectionné. Globalement nous pouvons distinguer deux types de méthodes pour l'extraction des correspondances entre les entités des ontologies :

- Les méthodes basées sur des optimisations locales (Euzenat et Valtchev (2004)): Le principe de la catégorie des méthodes locales consiste à extraire l'alignement final en itérant sur les correspondances formant l'alignement initial (typiquement la matrice de similarités), en maximisant localement la similarité à chaque paire d'entités ;

- Les méthodes basées sur une optimisation globale(Meillicke et Stuckenschmidt (2007)) : Contrairement aux méthodes locales, cette catégorie de méthodes consiste à optimiser un critère global. Typiquement on prend comme fonction objectif à maximiser la similarité globale des paires des entités correspondantes : $f = \sum_{C \in A} conf(C)$ où $conf(C) = conf < id_C, e_1, e_2, n, R >= n$ (voir la section 3.1)(Euzenat et Shvaiko (2013)). Le travail décrit dans (Meillicke et Stuckenschmidt (2015)) rentre dans cette catégorie et propose une approche probabiliste basée sur la logique de Markov pour modéliser le problème et comme critère à vérifier la maximisation de la probabilité de l'alignement extrait.

Les méthodes d'optimisation évoquées ci-dessus sont particulièrement adaptées pour extraire des alignements où la propriété d'injectivité (one-to-one alignment) est requise. Par contre pour l'extraction des alignements avec des cardinalités multiples (one-to-many or many-to-many) elles ne sont pas applicables. En outre,très peu de méthodes efficaces sont proposées dans la littérature. A notre connaissance le seul travail qui a abordé cette question est celui décrit dans (Cruz *et al*. (2009)).

3. Préliminaires sur l'alignement des ontologies

L'alignement entre deux ontologies est le processus de découverte des correspondances sémantiques entre les concepts de ces deux ontologies. Dans cette section nous présentons brièvement l'ensemble des concepts de base sur l'alignement des ontologies. Ces notions sont nécessaires pour la compréhension du contenu du papier. Pour plus de détails nous renvoyons le lecteur à la référence (Euzenat et Shvaiko (2013)).

3.1. Notion de correspondance

Soient O et O' deux ontologies. Une correspondance M entre O et O' est un quintuple $< id, e, e', R, n >$ où :

– id est un identificateur unique de la correspondance M ;

– e et e' sont des entités de O et O' respectivement (e.g., concepts, rôles ou instances) ;

– R est une relation sémantique entre e et e' (e.g., équivalence , plus général, plus spécifique, disjonction) ;

– n est une mesure de confiance, typiquement une valeur dans $[0, 1]$).

3.2. Notion d'alignement

L'alignement peut être défini comme un ensemble de correspondances. Le Processus d'alignement (figure 2) reçoit en entrée deux ontologies O et O' et produit en sortie un alignement A' entre les entités de O et O'. D'autres éléments viennent compléter cette définition, à savoir :

1. un alignement initial A à compléter ou à affiner par le processus ;
2. des ressources externes r à prendre en charge tel un thesaurus par exemple ;
3. des paramètres p tels que par exemple des seuils ou des poids.

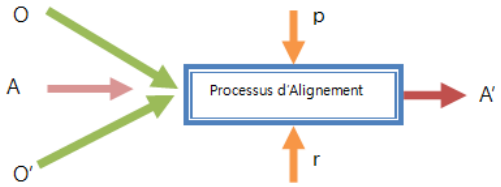


Figure 2. Processus d'alignement

3.3. Les contraintes de cardinalités et l'alignement des ontologies

A l'issue du processus de sélection, l'alignement résultat doit posséder les deux propriétés suivantes :

1. La similarité globale doit être maximale. La similarité globale désigne, la somme des valeurs des similarités des différentes correspondances qui forment l'alignement.

2. Les contraintes de cardinalités doivent être vérifiées. Nous distinguons en général les cas suivants :

- Cas 1 : la contrainte 1-1 : chaque entité de l'ontologie source doit correspondre à au plus une entité de l'ontologie cible et chaque entité de l'ontologie cible doit correspondre à au plus une entité de l'ontologie source ;

- Cas 2 : la contrainte n-m : chaque entité de l'ontologie source doit correspondre à au plus m entités de l'ontologie cible et chaque entité de l'ontologie cible doit correspondre à au plus n entités de l'ontologie source ;

- Cas 3 : Les contraintes n-*, *-m, *-* : dans ce cas on utilise le symbole * pour signifier qu'on n'impose pas de contrainte de cardinalité ;

Nous proposons dans ce papier, un modèle à base de flots qui permet d'extraire un alignement qui possède les deux propriétés ci-dessus. Ce modèle peut être faci-

lement adapté pour appréhender le cas d'un alignement total et injectif et considérer des contraintes de cardinalités plus générales où une entité d'une ontologie donnée (source ou cible) doit correspondre à au plus n entités et à au moins m entités de l'autre ontologie.

3.4. Mesures de similarité

Les différentes mesures de similarité utilisées dans le processus d'alignement sont organisées selon la classification suivante (Rahm et Bernstein (2001)) :

– la méthode terminologique : compare les labels des entités. Elle est décomposée en approches purement syntaxiques et celles utilisant un lexique. L'approche syntaxique effectue la correspondance à travers les mesures de dis similarité des chaînes (e.g. Distance d'édition). Tandis que, l'approche lexicale effectue la correspondance à travers les relations lexicales (e.g., synonymie, hyponymie, etc.) ;

– la méthode de comparaison des structures internes : compare les structures internes des entités (e.g., intervalle de valeurs, cardinalité d'attributs, etc.) ;

– la méthode de comparaison des structures externes : compare les relations d'entités avec d'autres. Elle est décomposée en méthodes de comparaison des entités au sein de leurs taxonomies et méthodes de comparaison des structures externes en tenant compte des cycles ;

– la méthode de comparaison des instances : compare les extensions des entités, i.e., elle compare l'ensemble des autres entités qui lui sont attachées (instances des classes) ;

– la méthode sémantique : compare les interprétations (ou plus exactement les modèles) des entités.

4. Algorithmes utilisés dans notre approche

Dans cette section, nous décrivons les différents algorithmes utilisés dans notre approche pour résoudre le problème d'extraction d'un alignement entre deux ontologies. Le premier algorithme est celui de Karp (Karp (1980)) qui est utilisé dans le cas des contraintes de cardinalités de types n-m, le deuxième est l'algorithme Hongrois (Kuhn (1955)) qui est utilisé dans le cas des contraintes de cardinalités de type 1-1. Et finalement nous présentons l'algorithme utilisé dans notre approche à savoir l'algorithme de flot de coût minimum (Ford et Fulkerson (1962); Liu (2003)).

4.1. Algorithme de Karp

L'algorithme de Karp (1980) fournit une solution efficace au problème d'extraction d'alignement. Pour faire une comparaison entre notre approche et cet algorithme, nous avons implémenté ce dernier en utilisant la stratégie de files d'attente prioritaires. Nous donnons ci-dessous une présentation globale du principe de cet algorithme.

Une instance du problème d'affectation est spécifiée par un graphe biparti non orienté complet, avec un coût non négatif associé à chaque arc. Le graphe est dénoté par $G = (V, E)$, où l'ensemble des sommets V est l'union de deux ensembles disjoints, X (les sources) et Y (les destinations), de telle sorte que $|x| \leq |y|$ ($|A|$ désigne le cardinal de l'ensemble A) ; un arc e est désigné par une paire $x \in X$ et $y \in Y$. Le coût sur l'arc $e = (x, y)$ est dénoté par $c(e)$ ou $c(x, y)$. Un couplage (Matching) dans G est un ensemble $M \subset E$ de telle sorte que chaque sommet est incident avec au plus une arête dans M. Le coût de M, est dénoté par $c(M)$, et est égal à $\sum_{e \in M} c(e)$ .Un couplage M est complet si chaque source est incident avec un certain arc dans M. La solution à ce problème d'affectation consiste à trouver un couplage complet de coût minimum.

Soit un couplage M, un sommet v est dit libre s'il n'est incident à aucun arc de M. un chemin dans G est dit alterné si ses arcs sont alternativement dans M. Un chemin alterné simple entre les sommets libres est dit un chemin augmentant. Si P est un chemin augmentant alors son sommet de début est une source et son sommet de fin est une destination. Si M est un couplage et P est un chemin augmentant alors la différence symétrique entre ces deux est aussi un couplage, et $|M \oplus P| = |M| + 1$. Le coût du chemin augmentant P est $c(M \oplus P) - c(M)$, qui peut être exprimé par

$$\sum_{e \in P - M} c(e) - \sum_{e \in P \cap M} c(e) \tag{1}$$

Le lemme bien connu suivant (Ford et Fulkerson (1962)) est fondamental :

LEMMA 1. — *Si M est un couplage de coût minimum de cardinalité k et P est un chemin augmentant de coût minimum relatif à M, alors $M \oplus P$ est le couplage de coût minimum de cardinalité $k + 1$.*

Le lemme 1 ci-dessus est la base de l'algorithme de Karp pour résoudre le problème d'affectation.

4.2. Algorithme Hongrois

Nous présentons dans cette section l'algorithme Hongrois qui a été utilisé dans d'autres travaux (Meilicke et Stuckenschmidt (2007); Chondrogiannis *et al.* (2014)) pour résoudre le problème de l'extraction d'un alignement entre deux ontologies.

L'algorithme Hongrois (Kuhn (1955)) résout le problème d'affectation en temps polynomial $O(n^4)$, mais Edmonds et Karp (1972) ont remarqué qu'il peut être modifié pour obtenir un $O(n^3)$.

Le problème résolu par l'algorithme hongrois s'énonce dans sa formulation initiale comme suit : Soit une matrice non-négative $n \times n$, où l'élément dans la $i^m e$ ligne et la $j^m e$ colonne représente le coût de l'attribution de la tâche j au travailleur i. Nous devons trouver une affectation des emplois aux travailleurs qui a un coût minimum. Si le but est de trouver la mission qui donne le coût maximum, le problème peut être modifié en soustrayant chaque coût du coût maximum. Cet algorithme, sert

à résoudre les problèmes d'affectation, considérant une matrice (appelée tableau de coûts), en choisissant un seul élément par ligne et par colonne de façon à rendre la somme minimale.

L'algorithme Hongrois fonctionne seulement sur des matrices carrées, et pour l'adapter et le rendre applicable sur m'importe quel type de matrice, nous avons utilisé l'approche proposée dans (Meillicke et Stuckenschmidt (2007)). Pour utiliser la méthode hongroise la matrice d'entrée M' doit être transformée en une matrice H. Chaque concept de l'ontologie source correspond à une ligne et chaque concept de l'ontologie cible correspond à une colonne. Puisque la méthode hongroise trouve une affectation minimale, une entrée dans la matrice doit être interprétée comme une distance entre deux concepts, où la distance entre un concept C_1 et un concept D_1 est définie à $1 - confiance(C1, D1, \equiv, c)$. Sans perte de généralité, on suppose que les valeurs de confiance d'entrée sont dans l'intervalle $[0, 1]$. Lorsqu'il n'existe pas une telle correspondance dans M' la distance est définie à ∞. Dans la plupart des situations d'adaptation, il ne sera pas possible de faire correspondre la totalité ou même la majorité des concepts. Par conséquent, la matrice d'entrée doit être étendue par des concepts supplémentaires qui jouent le rôle des candidats correspondants alternatifs. Nous appelons ces concepts les concepts fantômes. Ainsi, si n est le nombre de concepts de l'ontologie source T_1 et m est le nombre de concepts de l'ontologie cible T_2, on ajoute m lignes à la matrice d'entrée correspondantes aux m concepts fantômes, ainsi que n colonnes correspondantes aux n concepts fantômes. La valeur des entrées dans ces lignes (respectivement les colonnes) est définie à $1 + \varepsilon$ avec $\varepsilon >$ 0.

4.3. Algorithme de flot

Dans cette section nous décrivons l'algorithme du flot de coût minimum utilisé dans notre approche pour résoudre le problème de l'extraction de l'alignement entre deux ontologies. Le problème de flot de coût minimum est une généralisation du problème de flot maximum. Il est l'un des problèmes de flot les plus fondamentaux. Supposons que nous avons un réseau $G(V, E)$ avec des noeuds $V = 1, .., n$ et des arcs dirigés $E = (i, j) \in V \times V$. Le réseau G a deux noeuds spéciaux s et t nommés la source et le puits respectivement. Pour chaque arc dirigé $(i, j) \in E$, le coût pour passer une unité de flot du noeud i au noeud j est $c(i, j)$, et la capacité positive est $u(i, j)$. Le problème de flot de coût minimum consiste à trouver le flot maximum de coût minimum du noeud source s au noeud puits t.

Différentes approches sont proposées pour résoudre le problème de flot de coût minimum. Plusieurs discussions sur ce problème et son application peuvent être trouvées dans le livre et le papier de Ford et Fulkerson (1962), Edmonds et Karp (1972). Soit un réseau de flot $G(V, E)$ avec la source $s \in V$ et le puits $t \in V$, où un arc $(i, j) \in E$ a une capacité maximale $u(i, j)$, une capacité minimale $l(i, j)$, un flot $f(i, j)$ et un coût $c(i, j)$. Le coût de l'envoi de ce flot est $f(i, j) * c(i, j)$. La quantité de flot envoyée de s à t est d.

Le problème de flot à coût minimum consiste à déterminer comment acheminer dans les arcs du réseau une quantité v de flot d'une source $s \in V$ à une destination $t \in V$ de sorte à minimiser le coût total.

La définition du problème est de minimiser le coût total du flot : $\sum_{(i,j)\in V} c(i,j) * f(i,j)$ avec les contraintes :

$$\text{Contrainte de capacité :} l(i,j) \leq f(i,j) \leq u(i,j) \tag{2}$$

$$\text{Contrainte de symétrie :} f(i,j) = -f(j,i) \tag{3}$$

$$\text{Conservation de flot :} \sum_{w\in V} f(i,w) = 0 Ai \neq s,t \tag{4}$$

$$\text{Flot nécessaire :} \sum_{w\in V} f(s,w) = \sum_{w\in V} f(w,t) \tag{5}$$

Ce qui caractérise les problèmes de flots sont les contraintes de conservation de flot associées aux sommets du réseau. La contrainte de conservation de flot associée à un sommet i indique que la quantité totale de flot entrant dans le sommet doit être égale à celle sortant du sommet. L'algorithme s'exécute en temps pseudo polynomial. Cependant, supposons que les coûts $c(i,j)$ sont des entiers, et qui sont inférieurs ou égaux à un entier C, Edmonds and Karp (1972) ont prouvé que l'algorithme s'arrête après au maximum $1 + (1/4)(n^3 - n)(n-1)C$ augmentations de flot, qui est équivalant à $O(n^4C)$. Pour plus de détails sur l'algorithme nous renvoyons le lecteur à la référence (Liu (2003)).

5. Contribution

La contribution de ce papier porte sur le problème de l'extraction d'un alignement entre deux ontologies. Plus précisément, nous proposons une approche qui permet de sélectionner un alignement final ayant les deux propriétés suivantes :

1. l'alignement sélectionné est optimal, l'optimalité est mesurée à l'aide de la similarité globale entre les entités des deux ontologies à aligner. La similarité globale est définie comme la somme des similarités de toutes les correspondances sémantiques qui forment l'alignement.

2. l'alignement sélectionné vérifie les multiple contraintes de cardinalités spécifiées.

L'approche proposée est fondée sur la théorie des flots. Autrement dit, nous proposons de modéliser le problème de la sélection d'un alignement final comme un réseau de flots sur lequel nous appliquons l'algorithme de flot de coût minimum. Le flot de coût minimum obtenu assure l'optimalité de la fonction objectif qui représente la similarité globale et vérifie les contraintes de capacités qui représentent les contraintes de cardinalités multiple spécifiées. Nous décrivons ci-dessous le modèle proposé et la plateforme informatique développée pour son évaluation expérimentale.

Nous présentons dans cette section notre contribution en termes de modélisations du problème d'extraction d'un alignement entre deux ontologies. Nous présentons successivement notre modèle basé sur les flots et le modèle sous forme d'un problème d'affectation à résoudre par l'algorithme de Karp et l'algorithme Hongrois.

5.1. Modélisation de l'algorithme basé sur la théorie de flot

Nous présentons dans cette section notre contribution. Elle consiste en la construction d'un réseau qui modélise les contraintes de cardinalités avec un choix judicieux des contraintes de capacités. Ensuite, le choix des coûts et la recherche du flot de coût minimum qui assure l'optimalité de la similarité globale. Nous détaillons ci-dessous les règles de construction de ce réseau.

– Orienter chaque arc de chaque concept de l'ontologie O' à un concept de l'ontologie O. Pour un tel arc (u, v) : les bornes inférieure et supérieure des capacités sont initialisées comme suit : $l_{uv} = 0$ et $u_{uv} = 1$. Le coût associé à l'arc (u, v) est le suivant : $c(u, v) = 1 - sim(u, v)$;

– Ajouter un sommet s et un arc de s à chaque concept de l'ontologie O'. Pour un tel arc (s, a_i) : les bornes de capacités sont définies comme suit : $l_{sa_i} = 0$, $u_{sa_i} = n$. Le coût associé à chaque arc (s, a_i) est égal à 0. La valeur n représente la cardinalité de l'ontologie source O ;

– Ajouter un sommet t et un arc de chaque concept de l'ontologie O à t. Pour un tel arc (x, t) : les capacités limites sont définies comme suit : $l_{xt} = 0$, $u_{xt} = m$. Le coût associé à chaque arc (x, t) est égal à 0. La valeur m représente la cardinalité à l' ontologie cible O' ;

– Ajouter un arc (t, s) avec $l_{ts} = 0$. Le coût associé à l'arc (t, s) est égal à 0 et $u_{ts} = m\times$ (le nombre de concepts de l'ontologie O).

Nous remarquons que ce réseau permet de modéliser toutes les contraintes de cardinalité. Un alignement injectif, par exemple, peut être obtenu en fixant les bornes supérieures de capacités à 1. La propriété de complétude est assurée si nous fixons la borne inférieure de capacités de tous les arcs (x, t) à 1. Par ailleurs, ce modèle permet de tenir compte des contraintes de cardinalités plus générales. En effet, il est possible de représenter la contrainte suivante notamment : "chaque entité d'une ontologie peut être associée à au plus n entités et à au moins m autres entités de l'autre ontologie".

5.1.1. Exemple

Soient O et O' deux ontologies. O contient les concepts $\{a, b, c, d\}$ et O' contient les concepts $\{e, f, g, h, i\}$. Nous supposons qu'une certaine technique calculant les similarités entre les concepts de O et O' a produit la matrice de similarités S suivante (tableau 1). Nous supposons dans cet exemple que la cardinalité à l'ontologie source O est égale à 3 et que la cardinalité à l'ontologie cible O' est égale à 2.

Tableau 1. La matrice de similarité : S

O O'	e : 2	f : 2	g : 2	h : 2	i : 2
a : 3	0,81	0,61	0,73	0,61	0,50
b: 3	0,92	0,83	0,39	0,52	0,84
c : 3	0,64	0,62	0,26	0,74	0,94
d : 3	0,23	0,96	0,32	0,25	0,60

Nous supposons aussi que toutes les valeurs de similarités doivent être supérieures ou égales à un certain seuil $s = 0.6$ (voir la figure 3). Les correspondances inexactes au sens logique (criss-cross par exemple) ne sont pas prises en compte dans notre cas.

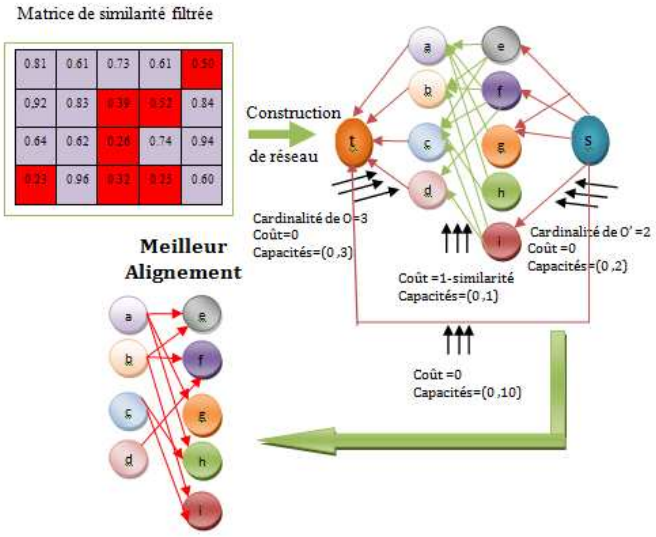


Figure 3. Le réseau de flot de l'exemple

Pour extraire le meilleur alignement en utilisant notre approche, nous avons besoin de construire le réseau de flots, après l'application de l'algorithme de calcul du flot de maximum de coût minimum, nous avons obtenu l'alignement présenté dans la figure 3, avec une similarité globale égale à $7,38$ (coût(a, e)+coût (a, g)+ coût (a, h)+ coût (b, e)+ coût (b, f)+ coût(b, i)+ coût(c, h)+coût (c, i)+coût (d, f)), c'est la contribution de notre papier : fournir le meilleur alignement qui maximise la similarité globale et qui vérifie les contraintes de cardinalités.

5.2. Modélisation sous forme d'un problème d'affectation

Dans cette section, nous modélisons le problème de l'extraction d'un alignement comme un problème d'affectation. Ensuite, nous exploitons deux algorithmes à savoir l'algorithme Hongrois et l'algorithme de Karp.

Soient O et O', deux ontologies ayant respectivement les concepts suivants $\{C_1, .., C_n\}$ et $\{C'_1, ..C'_m\}$. Notons que $sim(C_i, C'_j)$ désigne la similarité entre les concepts C_i et C'_j de l'ontologie O et l'ontologie O'. le problème de la sélection d'un alignement peut être modélisé comme un problème d'affectation spécifié à l'aide d'un graphe biparti $G(X, Y, E)$ comme suit :

– l'ensemble de sommets X, désigne l'ensemble des concepts $\{C_1, ..C_n\}$ de l'ontologie source O,

– l'ensemble de sommets Y, désigne l'ensemble des concepts $\{C'_1, ..C'_m\}$ de l'ontologie cible O',

– l'ensemble des arcs E, désigne l'ensemble des pairs (C_i, C'_j). L'arc (C_i, C'_j) est étiqueté avec la valeur de la similarité entre les concepts C_i et C'_j notée par $sim(C_i, C'_j)$.

5.3. Plateforme informatique

Afin de mettre en œuvre notre approche et de réaliser l'analyse expérimentale, nous avons développé une plateforme informatique dont l'architecture et le fonctionnement sont décrits ci-dessous.

La plateforme (voir la figure 4) peut être représentée comme un composant logiciel qui reçoit en entrée deux ontologies à aligner et fournit en sortie l'alignement entre les entités de ces deux ontologies. Notre architecture permet une composition parallèle où, plusieurs méthodes peuvent être utilisées sur la même entrée et ensuite combinées.

– L'étape d'analyse est la première étape ; ce composant permet d'analyser les deux ontologies à aligner qu'il reçoit en entrée. Cette analyse consiste à extraire les concepts des deux ontologies qui seront fournis sous forme d'une matrice vide dont les lignes représentent les concepts de l'ontologie source et les colonnes les concepts de l'ontologie cible.

– Le module de calcul des similarités : ce composant permet de calculer les similarités entre les entités des deux ontologies à aligner. Il fournit en sortie une matrice qui contient les similarités entre les entités de ces dernières. La majorité des systèmes d'alignement d'ontologies utilisent une combinaison de différentes mesures de similarité ; parce qu'il n'y a pas une mesure de similarité universellement pertinente pour toutes les applications. L'utilisation de plusieurs mesures de similarité donne souvent de meilleurs résultats. Le choix d'une ou de plusieurs mesures de similarité est paramétrable par l'utilisateur. Dans notre cas nous avons décidé d'utiliser deux types de similarités (les méthodes basées sur les chaînes : la distance d'édition, la distance de Jaro, la distance des n-gram et les méthodes basées sur une ressource : WordNet). Les méthodes choisies sont souvent les plus utilisées dans les différents systèmes, le choix de ces méthodes n'a pas fait l'objet d'attention particulière dans ce papier car l'objectif majeur est l'extraction d'un alignement optimal au sens de la performance en temps d'exécution. en effet selon cette métrique la performance de notre approche ne dépend pas de la qualité des similarités calculées. Bien entendu selon les autres

métriques (Précision, Rappel et f-Mesure) la qualité des similarités influe significativement sur l'alignement final extrait.

– Module d'agrégation des similarités : Ce composant permet de combiner les différentes valeurs de similarités calculées par les matchers cités avant. Les stratégies de combinaisons les plus fréquemment utilisées sont :

a) Le max : dans cette stratégie, on sélectionne la plus grande valeur de similarité.

b) Le min : dans cette stratégie, on sélectionne la plus petite valeur de similarité.

c) La moyenne : dans cette stratégie, on calcule la moyenne des valeurs de similarité.

d) La moyenne pondérée : dans cette stratégie, on calcule la moyenne pondérée des valeurs de similarité. A ce niveau, nous avons deux options. La première est la saisie manuelle des poids (le poids est saisi par l'utilisateur) et la deuxième est le calcul automatique des poids (le poids est généré automatiquement par différentes méthodes). Plusieurs études ont abordé le problème de l'estimation des poids des différents matchers. Dans (Wang *et al.* (2010)), les auteurs proposent une approche basée sur la théorie de l'information et d'estimer le poids de chaque matcher en se basant sur le calcul de l'entropie (incertitude des informations) à partir des valeurs de similarité calculées par ce Matcher. Les travaux décrits dans (Martinez-Gil *et al*. (2008)) et (Wang *et al.* (2006)) présentent une approche basée sur des algorithmes génétiques pour donner une estimation des poids attribués aux différentes stratégies utilisées. Dans (Mao *et al*. (2008)), les auteurs proposent le concept de l'harmonie pour le pesage des différents matchers. D'autres travaux tel que (Ichise (2008)) utilisent des techniques d'apprentissage pour la configuration automatique des poids à attribuer aux matchers.

Dans notre cas, nous avons utilisé la stratégie d'agrégation, la moyenne pondérée mentionnée ci-dessus pour réaliser l'étape d'agrégation avec un poids égal à 1 affecté à chaque matcher.

– Le module de filtrage : la matrice de similarité finale, qui est la sortie de l'étape d'agrégation, devient l'entrée du module de filtrage. Ce module permet de détecter d'une manière automatique les relations des correspondances non pertinentes. Il s'agit d'une opération de filtrage des correspondances qui sera réalisée à l'aide d'un seuil introduit par l'utilisateur du système.

– La dernière étape est la sélection des correspondances ; cette étape prend comme entrée soit une matrice de similarité calculée à partir de deux ontologies réelles obtenue après l'exécution des étapes : d'analyse, de calcul de similarité, d'agrégation et de filtrage, soit une matrice de similarité générée de manière aléatoire (données synthétiques). Dans l'étape de sélection des correspondances, nous avons deux opérations à effectuer, la première étape est de construire un réseau de flot afin d'exécuter l'algorithme de flot de coût minimum utilisé dans notre approche et la deuxième étape consiste à construire un graphe biparti pour exécuter l'algorithme Karp et l'algorithme hongrois ; Ces algorithmes sont utilisés pour faire des comparaisons avec

notre approche. Après l'exécution de ces trois algorithmes sur différentes matrices de similarité et avec différentes contraintes de cardinalité, nous obtenons trois résultats ; chacun représente le meilleur alignement trouvé par l'algorithme utilisé.

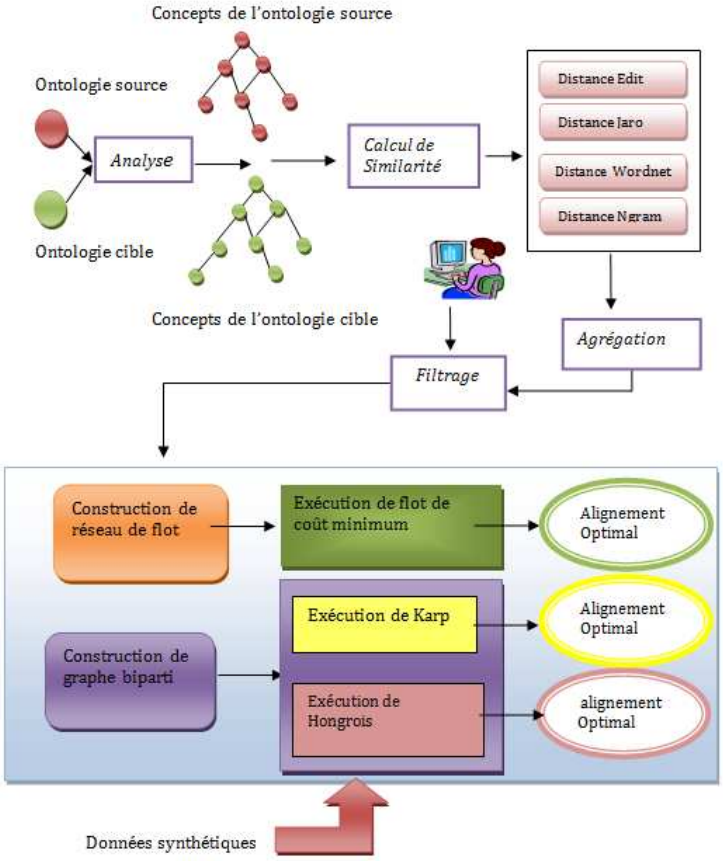


Figure 4. L'architecture de notre plateforme

6. Expérimentation et discussion

Nous détaillons dans cette section nos résultats expérimentaux. En effet, nous n'avons pas trouvé dans la littérature spécialisée d'autres systèmes ayant abordé le problème de l'extraction d'un alignement avec des contraintes de cardinalités multiples et ayant fourni des résultats détaillés pouvant être utilisés comme support de comparaison avec notre approche à l'exception du travail décrit dans (Cruz *et al.* (2009)). Toutefois ce dernier ne fournit des résultats détaillés que pour les matrices

carrées. Pour cette raison, nous avons implémenté l'algorithme de Karp (1980) utilisé dans ce travail pour être en mesure de comparer les deux approches.

Nous utilisons l'approche automatique et la méthode basée sur l'optimisation globale. Notre objectif n'est pas d'extraire juste un alignement, mais d'en extraire le meilleur alignement qui maximise la fonction objectif et qui vérifie en même temps les contraintes de cardinalités.

L'implémentation de notre application est divisée en deux parties, la première partie qui consiste à effectuer les étapes d'analyse et de calcul de la matrice de similarité finale a été réalisée avec le langage de programmation JAVA sous la plateforme NetBeans IDE 7.0.1. La deuxième partie qui consiste à construire le réseau de flot et d'exécuter l'algorithme de flot de coût minimum à été réalisée avec le langage C++. Le temps d'exécution mentionné dans les graphes a été obtenu avec le langage C++. Notons que nous avons utilisé une machine de 4 Giga octet de RAM et un processeur Intel Core i5 de vitesse 2.5 Ghz.

Le critère d'évaluation considéré dans nos expérimentations pour mesurer la performance de notre approche est le temps d'exécution.

Pour évaluer la performance de notre approche nous avons réalisé une étude empirique. Dans cette analyse expérimentale, nous avons considéré les scénarios suivants :

1. Scénario 1 : nous avons comparé la performance de notre approche avec celle de l'algorithme de Karp sur des matrices carrées et rectangulaires pour des contraintes de cardinalités de type $(1-1)$ et $(n-m)$.

2. Scénario 2 : nous avons comparé la performance de notre approche avec l'algorithme hongrois sur des matrices carrées et rectangulaires pour des contraintes de cardinalités de type $(1-1)$.

Nous avons considéré deux types de données :

1. Les données synthétiques : ces données sont générées aléatoirement. Elles ont l'avantage de permettre la génération de cas de test qui soient ciblés et diversifiés.

2. Les données réelles (ontologies de domaine), ce type de données est utilisé pour examiner le comportement des algorithmes dans des cas réels.

6.1. Comparaison avec l'algorithme de Karp (données synthétiques)

Dans cette section, nous présentons les résultats de la comparaison entre l'algorithme de flot de coût minimum utilisé dans notre approche et l'algorithme de Karp. (Les contraintes de cardinalité prises en compte dans ce cas sont de type $(n-m)$).

Nous avons noté trois principaux résultats de nos expérimentations.

– Cas $1-1$: matrices de similarité rectangulaires, l'algorithme du flot de coût minimum et l'algorithme de Karp retournent des résultats dans le même temps d'exécution. Par exemple avec une certaine matrice $100 \times 1\,000$ notre approche donne une

solution après 1 seconde et l'algorithme de Karp donne une solution après $0,5$ secondes (voir la figure 5). Mais sur les matrices carrées, l'algorithme de Karp est plus performant que notre algorithme.

– Cas $n-m$: matrices de similarité rectangulaires, par exemple pour une certaine matrice de $200 \times 2\ 000$ avec des contraintes de cardinalité égale à 4 pour l'ontologie source et 3 pour l'ontologie cible, l'algorithme du flot de coût minimum donne une solution au bout de 11 secondes, alors que l'algorithme de Karp donne une solution après 28 secondes. L'interprétation de ce résultat est que, notre algorithme réagit directement sur la matrice telle qu'elle, alors que l'algorithme de Karp est réitéré plusieurs fois de manière séquentielle pour vérifier les contraintes de cardinalité et retourner ce résultat. En outre, notre approche donne de meilleurs résultats pour les ontologies où la différence entre le nombre de concepts des deux ontologies à aligner est importante (c-à-d lorsque le facteur de rectangularité RF est inférieur strictement à $0,5$ ($RF = 200/2\ 000 = 0,1$)). Alors que dans le cas de grandes ontologies et de faible variation(c-à-d les deux ontologies ont presque le même nombre de concepts (R proche de 1)), les résultats sont moins bons que ceux de l'algorithme de Karp (voir la figure 6). Concernant les matrices carrées l'algorithme de Karp donne de meilleurs résultats que l'algorithme du flot à coût minimum.

– L'efficacité de notre algorithme dépend des contraintes de cardinalité : Si nous changeons les contraintes de cardinalité nous obtenons les résultats présentés dans la figure 7. Avec les contraintes de cardinalités égales à 20 pour l'ontologie source et 10 pour l'ontologie cible, l'algorithme réagit plus lentement que si nous utilisons l'algorithme avec des contraintes de cardinalités égales à 4 pour l'ontologie source et 3 pour l'ontologie cible.

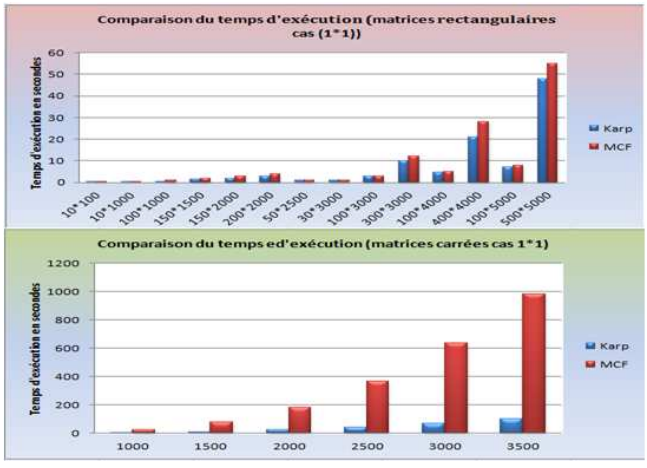


Figure 5. Comparaison entre l'algorithme de Karp et l'algorithme du flot de coût minimum cas 1-1(Matrices rectangulaires et carrées)

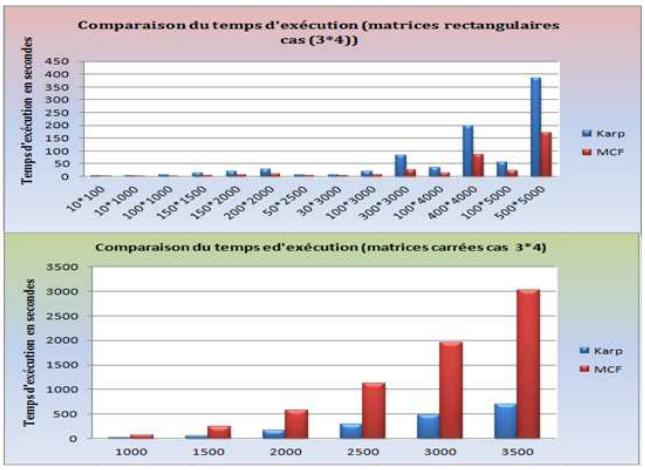


Figure 6. Comparaison entre l'algorithme de Karp et l'algorithme du flot de coût minimum cas 3-4(Matrices rectangulaires et carrées)

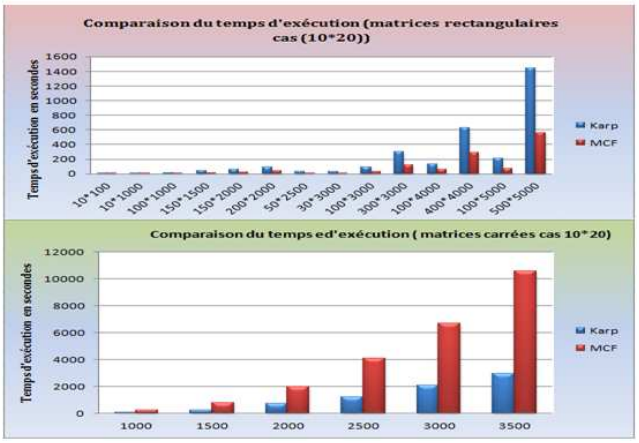


Figure 7. Comparaison entre l'algorithme de Karp et l'algorithme du flot de coût minimum cas 10-20(Matrices rectangulaires et carrées)

6.2. Comparaison avec l'algorithme de Karp (données réelles)

Afin d'évaluer notre approche sur des données réelles, nous avons utilisé deux ensembles de données publiés dans le site OAEI 2010 [1], il s'agit des l'ensembles de données Anatomy , Directory et l'ensemble LargeBio de la campagne de OAEI 2015.

1. http://oaei.ontologymatching.org/

Depuis 2005, la base de données originale du site OAEI a été légèrement modifiée, les deux campagnes de 2011 et 2012 contenaient d'autres ensembles de données qui ont été générés automatiquement(Rosoiu *et al*. (2011)).

6.2.1. L'ensemble de données Anatomy

Le cas de test "anatomy" consiste à trouver un alignement entre l'ontologie "NCI Thesaurus" (Hayamizu *et al*. (2005)) qui décrit l'anatomie humaine (contient 2 774 classes) et l'ontologie "Adult Mouse" (Sioutos *et al*. (2007)) qui décrit l'anatomie des souris (contient 3 304 classes). Ces deux ressources font partie d'OBO (Open Biomedical Ontologies). L'alignement entre ces ontologies a été créé par les experts du domaine (Bodenreider *et al*. (2005)). La tâche est placée dans une zone où il ya de grandes ontologies qui sont décrites en termes techniques.

Dans la figure 8, nous comparons les résultats renvoyés par l'algorithme de flot de coût minimum et l'algorithme de Karp sur l'ensemble de données "Anatomy". Nous pouvons noter les trois principales observations de cette figure :

– Les contraintes de cardinalité égales à $1-1$: l'algorithme de flot de coût Minimum donne une solution après 675 secondes et l'algorithme de Karp donne une solution après 294 secondes, alors nous pouvons dire que l'algorithme de Karp est meilleur que notre algorithme. Ce résultat est compatible avec les résultats mentionnés ci-dessus pour les données synthétiques.

– $2-2 \leq$ Les contraintes de cardinalité $\leq 3-4$: dans ce cas, nous observons que les résultats des deux algorithmes sont très proches parce qu'ils retournent les résultats après presque le même temps d'exécution, par exemple pour les contraintes de cardinalité égales à $3-3$, l'algorithme de flot à coût minimum donne une solution après 1 705 secondes et l'algorithme de Karp donne une solution après 1 764 secondes.

– Les contraintes de cardinalité > $3-4$: dans ce cas, l'algorithme de flot de coût minimum est meilleur que l'algorithme de Karp.

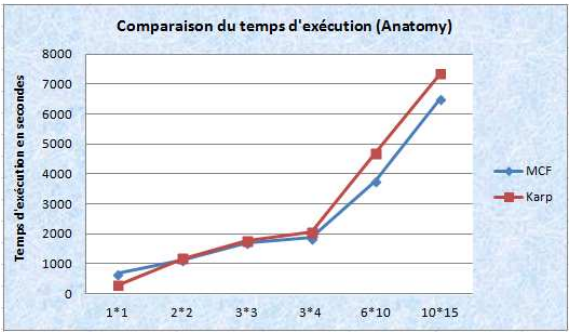


Figure 8. Comparaison entre l'algorithme de flot de coût minimum et l'algorithme de Karp sur l'ensemble de données "Anatomy" avec différentes contraintes de cardinalité

Ce qui caractérise ce cas de test est que les deux ontologies sont de type matrices rectangulaires (l'ontologie 1 : 2 744 classes et l'ontologie 2 : 3 304 classes) avec une différence de 560 classes, le facteur de rectangularité RF est égal à $0,83$ qui est supérieur strictement à $0,5$ ($RF = 2\ 744/3\ 304$), donc c'est pour cette raison, que les résultats obtenus pour l'ensemble de données d'anatomie ne sont pas très satisfaisants, ceci est donc compatible avec notre analyse concernant les données synthétiques donnée ci-dessus.

6.2.2. L'ensemble de données Directory

L'ensemble de données Directory, est le cas réel qui consiste à aligner les répertoires des sites web (comme Open Directory ou Yahoo). Cette base de données est constituée d'un seul test qui correspond à deux grands répertoires (l'ontologie 1 : 2 854 classes et l'ontologie 2 : 6 555 classes).

Dans la figure 9, nous comparons les résultats renvoyés par l'algorithme de flot de coût minimum et l'algorithme de Karp sur l'ensemble de données Directory. Nous pouvons noter les deux principales observations suivantes à partir de cette figure :

– Les contraintes de cardinalité égales à $1-1$: dans ce cas, les deux algorithmes ont presque le même temps d'exécution.

– Les contraintes de cardinalité> $1-1$: notre algorithme est plus performant que l'algorithme de Karp. Le facteur de rectangularité dans ce cas est égal à $0,43$, qui est inférieur à $0,5$ et comme nous l'avons déjà prouvé sur des données synthétiques, notre approche donne de bons résultats dans ce cas d'alignement (lorsque $RF < 0,5$, ce qui signifie que la différence entre le nombre de concepts est important) et ce cas montre très bien l'efficacité de notre algorithme.

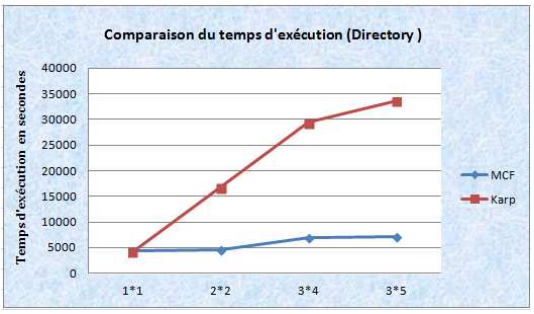


Figure 9. Comparaison entre l'algorithme de flot de coût minimum et l'algorithme de Karp sur l'ensemble de données "Directory" avec différentes contraintes de cardinalité

6.2.3. L'ensemble de données LargeBio

L'ensemble de données LargeBio consiste à trouver des alignements entre le modèle d'anatomie (FMA), SNOMED CT, et le Thesaurus de l'institut national de can-

cer (NCI). Ces ontologies sont sémantiquement riches et contiennent des dizaines de milliers de classes. Cet ensemble de données se compose de plusieurs tâches d'alignement impliquant différents fragments des ontologies FMA, NCI et SNOMED CT. Parmi l'ensemble de tâches nous avons pu exécuter la tâche suivante FMA-NCI petits fragments : Cette tâche consiste à associer deux petits fragments de FMA et NCI. Le fragment FMA contient 3 696 classes (5 % des FMA), tandis que le fragment NCI contient 6 488 classes (10 % du NCI). A noter que cette tâche est plus complexe (3 696 classes sur 6 488 classes) que la tâche relative à l'ensemble de données Anatomie décrite ci-dessus (2 774 classes sur 3 304 classes).

Dans la figure 10, nous comparons les résultats renvoyés par l'algorithme de flot de coût minimum et l'algorithme de Karp sur l'ensemble de données LargeBio.

Les résultats obtenus pour cette tâche de l'ensemble de données LargeBio montre que l'algorithme de flot de coût minimum est plus performant que l'algorithme de Karp, le facteur de rectangularité dans ce cas est égal à $0,56$, qui est proche de $0,5$ et comme nous avons déjà montré, notre algorithme retourne des résultats satisfaisants sur ce type d'ontologies.

Pour les autres tâches de l'ensemble LargeBio, dans sa globalité, notre approche ne donne pas des résultats en temps réel. Nous envisageons dans le futur d'étendre notre méthode pour un meilleur passage à l'échelle. Cette méthode semble efficace pour des ontologies dont le nombre de concepts ne dépasse pas 10 000 concepts pour chaque ontologie.

Figure 10. Comparaison entre l'algorithme de flot de coût minimum et l'algorithme de Karp sur l'ensemble de données "LargeBio" avec différentes contraintes de cardinalité

6.3. Comparaison avec l'algorithme Hongrois

Dans la figure 11, nous comparons les résultats de l'algorithme Hongrois [2] avec les résultats de l'algorithme de flot de coût minimum utilisé dans notre approche.

Nous avons noté deux résultats principaux de nos expérimentations :

– Matrices de similarités rectangulaires : notre algorithme de flot de coût minimum traite efficacement le problème par rapport à l'algorithme Hongrois. Par exemple sur une matrice $100 \times 1\ 000$ l'algorithme de flot de coût minimum a donné une solution après 12 secondes alors que l'algorithme Hongrois a donné une solution après 38 secondes. L'une des raisons qui expliquent ce comportement, est que l'algorithme Hongrois transforme la matrice en une matrice $1\ 100 \times 1\ 100$, alors que l'algorithme de flot de coût minimum agit directement sur la matrice sans la transformer. Donc pour les matrices de grandes dimensions l'algorithme Hongrois est moins performant que l'algorithme basé sur les flots.

– Matrices de similarités carrées : L'algorithme Hongrois est meilleur. Puisque cet algorithme est établi dans la pratique pour trouver une affectation optimale sur ce type de matrices.

Enfin, on peut conclure que l'algorithme de calcul du flot à coût minimum présente un avantage certain par rapport à l'algorithme Hongrois dans le contexte de l'alignement des ontologies. En effet, les ontologies correspondent généralement à des matrices rectangulaires et il est très rare d'avoir dans la réalité des ontologies à aligner avec le même nombre de concepts.

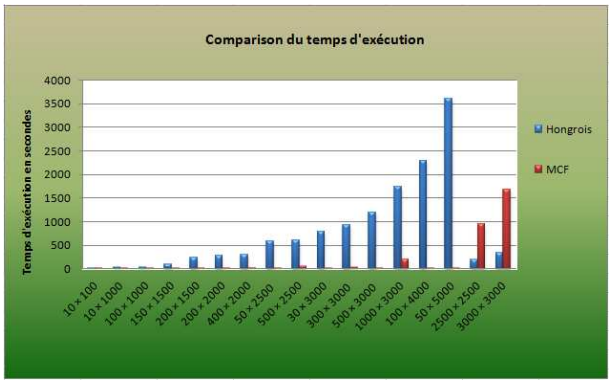


Figure 11. La représentation des résultats de l'algorithme Hongrois et l'algorithme de flot de coût minimum

En exploitant les complexités données de ces deux algorithmes (Hongrois $O(n^3)$, et l'algorithme de flot de coût minimum $O(n^4C)$), et puisque toutes les capacités sont

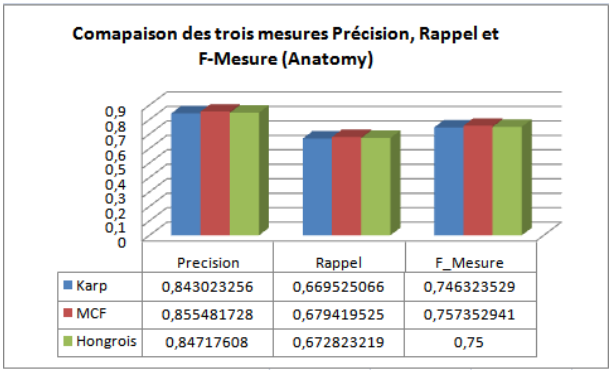
2. L'implémentation est disponible sur le site http://saebyn.info/2007/05/22/munkres-code-v2/

inférieures à 1, la complexité de l'algorithme de flot de coût minimum peut s'écrire sous $O(n^4)$. Donc, c'est prouvable que sur les matrices carrées, l'algorithme Hongrois est meilleur. Par contre il n'est pas préférable dans le cas contraire c.-à-d., sur les matrices rectangulaires ce qui est confirmé par notre expérimentation.

6.4. Autres mesures d'évaluations

Pour évaluer la qualité des alignements produits par les trois algorithmes utilisés dans notre approche (Flot de coût minimum, Karp et Hongrois) et pour comparer entre eux les résultats des processus d'alignement, nous utilisons les mesures habituelles : précision, rappel et F-Mesure adaptées à l'alignement d'ontologies. Ces mesures s'appuient sur une comparaison avec un alignement de référence.

La figure 12 présente les différents résultats trouvés pour les trois mesures précision, Rappel et F-Mesure des différents algorithmes utilisés dans notre approche sur l'ensemble de données "Anatomy". En analysant cette figure, nous remarquons que les trois algorithmes ont presque les mêmes valeurs pour les trois mesures Précision, rappel et F-Mesure.



	Precision	Rappel	F_Mesure
Karp	0,843023256	0,669525066	0,746323529
MCF	0,855481728	0,679419525	0,757352941
Hongrois	0,84717608	0,672823219	0,75

Figure 12. Comparaison entre l'algorithme de flot de coût minimum, l'algorithme de Karp et l'algorithme Hongrois sur l'ensemble de données "Anatomy"

7. Conclusion

Dans cet article, nous avons montré que le problème de l'extraction de l'alignement des ontologies peut bénéficier des techniques algorithmiques développées au sein de la théorie des flots. Plus précisément, nous avons modélisé le problème de l'extraction d'un alignement satisfaisant des contraintes de cardinalités et la fonction objectif définie comme la similarité globale entre les entités des ontologies comme un réseau de flot. Afin d'extraire un tel alignement, nous avons proposé un modèle basé sur les flots et pour évaluer notre approche, nous avons utilisé deux types de données (données synthétiques et données réelles). Dans notre expérimentation, nous avons

comparé notre algorithme avec les deux algorithmes les plus largement utilisés pour résoudre le problème de l'alignement d'ontologies : l'algorithme de Karp et l'algorithme Hongrois. Enfin, nous concluons que l'algorithme de flot de coût minimum présente un avantage incontestable par rapport à l'algorithme de Karp et l'algorithme Hongrois. En effet, les ontologies correspondent généralement à des matrices rectangulaires et il est très rare d'avoir dans la réalité des ontologies avec le même nombre de concepts.

Bibliographie

Bagher H., Abolhassani H. et Sayyadi H. (2006). A neural networks based approach for ontology alignment. In *The 3rd international conference on soft computing and intelligent systems and the 7th international symposium on advanced intelligent systems.*

Bellahsene Z., Bonifati A. et Rahm E. (2011). *Schema matching and mapping*. Springer.

Bodenreider O., Hayamizu T., Ringwald M., Coronado S. D. et Zhang S. (2005). Of mice and men: Aligning mouse and human anatomies. In *Proceedings of american medical informatics association (aima) annual symposium*, p. 61-65.

Chondrogiannis E., Andronikou V., Karanastasis E. et Varvarigou T. A. (2014). An intelligent ontology alignment tool dealing with complicated mismatches. In *Proceedings of the 7th international workshop on semantic web applications and tools for life sciences, berlin, germany, december 9-11, 2014.* Consulté sur http://ceur-ws.org/Vol-1320/paper_16.pdf

Cruz I., Antonelli F. et Stroe C. (2009). Efficient selection of mappings and automatic quality-driven combination of matching methods. In *International workshop on ontology matching*, p. 49-60.

David J., Guillet F. et Briand H. (2007). Association rule ontology matching approach. *International Journal of Semantic Web Information Systems*, vol. 3, n° 2, p. 27-49.

Do H. et Rahm E. (2007). Matching large schemas: Approaches and evaluation. *Information Systems*, vol. 32, n° 6, p. 857–885.

Eckert K., Meilicke C. et Stuckenschmidt H. (2009, June). Improving ontology matching using meta-level learning. In *The semantic web: Research and applications, 6th european semantic web conference, ESWC 2009, heraklion, crete, greece, may 31-june 4, 2009, proceedings*, p. 158-172.

Euzenat J. et Shvaiko P. (2013). *Ontology matching*. Springer.

Euzenat J. et Valtchev P. (2004). Similarity-based ontology alignment in OWL-lite. In *16th european conference on artificial intelligence (ecai)*, p. 333-337. IOS Press.

Ford L. R. et Fulkerson D. R. (1962). *Flows in networks*. Princeton University Press.

Giunchiglia F., Yatskevich M. et Shvaiko P. (2007). Semantic matching: Algorithms and implementation. *Journal of Data Semantics*, vol. 4604, p. 1-38.

Hayamizu T., Mangan M., Corradi J., Kadin J. A. et Ringwald M. (2005). The adult mouse anatomical dictionary: a tool for annotating and integrating data. *Genome Biology*. Consulté sur http://www.ncbi.nlm.nih.gov/pubmed/15774030?dopt=Abstract

Ichise R. (2008). Machine learning approach for ontology mapping using multiple concept similarity measures. In *7th IEEE/ACIS international conference on computer and information science, IEEE/ACIS ICIS 2008, 14-16 may 2008, portland, oregon, USA*, p. 340-346.

Jean-Mary Y., Shironoshita E. et Kabuka M. (2009). Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, n° 3.

Jiménez-Ruiz E., Grau B. C., Zhou Y. et Horrocks I. (2012). Large-scale interactive ontology matching: Algorithms and implementation. In *ECAI 2012 - 20th european conference on artificial intelligence. including prestigious applications of artificial intelligence (PAIS-2012)system demonstrations track, montpellier, france, august 27-31 , 2012*, p. 444-449.

Jiménez-Ruiz E. et Grau B. C. (2011). Logmap: Logic-based and scalable ontology matching. In *International semantic web conference, lncs*, vol. 7031, p. 273-288. Springer.

Kalfoglou Y. et Schorlemmer M. (2003, janvier). Ontology mapping: The state of the art. *The Knowledge Engineering Review*, vol. 18, n° 1, p. 1-31.

Karp R. M. (1980). An algorithm to solve the m * n assignment problem in expected time o(mn log n). *Networks*, p. 143-152.

Kengue F. D. J., Euzenat J. et Valtchev P. (2008). Alignement d'ontologies dirigé par la structure. In Y. A. Ameur (Ed.), *Cal*, vol. RNTI-L-2, p. 155. Cépadués-Éditions.

Kuhn H. w. (1955). The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, vol. 2, n° 1-2, p. 83-97.

Liu J. (2003). *Algorithms for minimum cost flows*. The University of Western Ontario.

Mao M., Peng Y. et Spring M. (2008). A harmony based adaptive ontology mapping approach. In *Proceedings of the 2008 international conference on semantic web web services, SWWS 2008, july 14-17, 2008, las vegas, nevada, usa*, p. 336-342.

Martinez-Gil J., Alba E. et Montes J. A. (2008). Optimizing ontology alignments by using genetic algorithms. In *Proceedings of the first international workshop on nature inspired reasoning for the semantic web, aachen, germany, october 27*, vol. 419, p. 1-15. CEUR-WS.org.

Meilicke C. et Stuckenschmidt H. (2007). Applying logical constraints to ontology matching. In *The 30th german conference on artificial intelligence*, vol. 4667, p. 99-113. Springer.

Meilicke C. et Stuckenschmidt H. (2015). New paradigm for ontology alignment. In *In the proceedings of the tenth international workshop on ontology matching, collocated with the 14th international semantic web conference, ceur-ws*, vol. 1545, p. 1-12. Bethlehem, PA, USA, CEUR-WS.org.

Noy N. (2004). Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*, vol. 33, n° 4, p. 65-70.

Noy N. et Musen M. A. (2002). The prompt suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, vol. 59, n° 6, p. 983-1024.

Rahm E. et Bernstein P. A. (2001, décembre). A survey of approaches to automatic schema matching. *The VLDB Journal*, vol. 10, n° 4, p. 334-350.

Rosoiu M. E., Santos C. T. dos et Euzenat J. (2011). Ontology matching benchmarks: generation and evaluation. In *6th iswc workshop on ontology matching (om)*, p. 73-84.

Shvaiko P. et Euzenat J. (2005a). A survey of schema-based matching approaches. *Journal on Data Semantics*, vol. 4, p. 146-171.

Shvaiko P. et Euzenat J. (2005b). *Tutorial on schema and ontology matching.* Consulté sur http://disi.unitn.it/~accord/Presentations/ESWC'05-MatchingHandOuts.pdf(le20/04/2014)

Shvaiko P. et Euzenat J. (2008). Ten challenges for ontology matching. In *Proceedings of the seventh international conference on ontologies, data bases and applications of semantics*, vol. 5332, p. 1164-1182. Berlin, Heidelberg, Springer-Verlag.

Sioutos N., Coronado S. de et Haber M. (2007). Nci thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, p. 30-43.

Xingsi X., Yuping W. et Weichen H. (2015). Optimizing ontology alignments by using nsga-ii. *International Arab Journal of Information Technology*, vol. 12, n° 2.

Zhang S. et Bodenreider O. (2007). Experience in aligning anatomical ontologies. *International Journal on Semantic Web and Information Systems*, vol. 3, n° 2, p. 1-26.