Similarité de graphes : une mesure générique et un algorithme tabou réactif

Sébastien Sorlin, Christine Solnon

Similarité de graphes : une mesure générique et un algorithme tabou réactif

Sébastien Sorlin et Christine Solnon

LIRIS, CNRS UMR 5205, bât. Nautibus, Université de Lyon I
43 Bd du 11 novembre, 69622 Villeurbanne cedex, France
{sebastien.sorlin,christine.solnon}@liris.cnrs.fr

Résumé :
De nombreuses applications nécessitent de mesurer la similarité d'objets, *e.g.*, la reconnaissance d'images, la recherche d'information, le raisonnement à partir de cas... Lorsque les objets sont représentés par des graphes, ce problème se ramène à mesurer la similarité de graphes. Différents types d'appariements de graphes, définissant chacun une mesure de similarité différente, ont été proposés : l'isomorphisme de (sous-)graphes, les appariements à tolérance d'erreur, la distance d'édition de graphes... Un premier objectif de cet article est de montrer que ces mesures peuvent être vues comme des cas particuliers d'une mesure de similarité introduite dans Champin & Solnon (2003). Initialement définie pour la comparaison d'objets de conception, cette mesure est basée sur un appariement multivoque de deux graphes (*i.e.*, un appariement où chaque sommet peut être apparié à plusieurs sommets) ce qui permet la comparaison d'objets décrits à des niveaux de granularité différents ou d'images sur- ou sous-segmentées. Nous nous intéressons ensuite au calcul de cette similarité et nous décrivons deux algorithmes : un algorithme glouton capable de calculer rapidement une approximation de la similarité de deux graphes et une recherche locale taboue réactive améliorant cette approximation. Quelques résultats expérimentaux sont donnés.

Mots-clés : Similarité, Appariement de graphes, Isomorphisme de graphes, Recherche locale, Méta-heuristique taboue

1 Introduction

Les graphes sont souvent utilisés pour modéliser des objets structurés. Dans Champin & Solnon (2003), des objets de conception sont modélisés par des graphes étiquetés où les sommets représentent les composants des objets et les arcs représentent les relations entre ces composants. Afi n de distinguer les différents types de composants et de relations, des étiquettes sont ajoutées aux sommets et aux arcs. Les graphes permettent également la représentation d'images (*e.g.*, Ambauen *et al.* (2003)) : un sommet représente une région de l'image et peut être étiqueté par ses propriétés (couleur, taille....) et un arc représente une relation binaire entre deux régions et peut être étiqueté par la nature de la relation (connectivité, position relative...).

Comparer deux objets revient alors à comparer deux graphes, *i.e.*, mettre en relation leurs sommets (les apparier) afi n d'identifi er leurs points communs. Cette comparaison peut se faire à travers la recherche d'une relation d'isomorphisme de (sous-)graphes afi n de montrer l'existence d'une relation d'équivalence ou d'inclusion entre les deux graphes. Cependant, deux objets "similaires" ne sont pas nécessairement "identiques" et présumer de l'existence d'une relation permettant de retrouver tous les sommets et tous les arcs est généralement utopique. Par conséquent, des techniques de comparaison de graphes à tolérance d'erreurs telles que la recherche du plus grand sous-graphe commun ou la "graph edit distance" ont été proposées (Bunke (1997); Conte *et al.* (2004)).

Plus récemment, trois articles différents proposent de franchir une étape supplémentaire dans la comparaison de deux graphes en introduisant la notion d'appariement multivoque, *i.e.*, un appariement permettant d'associer un sommet d'un graphe à un ensemble de sommets de l'autre graphe :

- Dans Champin & Solnon (2003), des graphes sont utilisés pour représenter des objets de conception. Dans ce contexte et selon le niveau de granularité de représentation des objets, un seul composant d'un objet peut jouer le rôle d'un ensemble de composants d'un autre objet. Pour en tenir compte, les auteurs utilisent des appariements multivoques permettant de relier un sommet d'un graphe à un ensemble de sommets de l'autre graphe.
- Dans Boeres *et al.* (2004), l'appariement de graphes est utilisé pour comparer des images segmentées du cerveau à un modèle du cerveau. Le modèle, d'aspect schématique, est "correctement" segmenté alors que les images, bruitées, sont généralement sur-segmentées. Il n'existe donc pas de bijection entre les régions du modèle et les régions de l'image. Les auteurs utilisent une mesure de similarité entre deux images basée sur des appariements multivoques où un sommet du schéma peut être apparié à plusieurs sommets de l'image.
- Dans Ambauen *et al.* (2003), une nouvelle distance d'édition de graphes introduisant deux nouvelles opérations –la fusion et l'éclatement de sommets– est proposée. Ces deux nouvelles opérations permettent de prendre en compte le fait que les images à comparer sont généralement sur- ou sous- segmentées.

Motivation et plan. Un premier objectif de cet article est de mettre en évidence les points communs entre les appariements proposés dans ces trois articles. Un second objectif est de proposer des algorithmes pour la recherche du meilleur appariement multivoque de deux graphes. La partie 2 introduit brièvement la mesure de similarité proposée par Champin & Solnon (2003). En section 3, on montre que cette mesure générique peut s'instancier en d'autres mesures de similarité. En section 4 nous nous intéressons au problème du calcul de cette similarité : nous proposons un algorithme glouton calculant rapidement une approximation de la similarité ainsi qu'une recherche locale taboue réactive améliorant les résultats obtenus par l'algorithme glouton. Finalement, la section 5 présente quelques résultats expérimentaux.

2 Une mesure de similarité de graphes multi-étiquetés

Un **graphe orienté** est défi ni par un couple $G = (V, E)$, où V est un ensemble fi ni de sommets et $E \subseteq V \times V$ est un ensemble fi ni d'arcs orientés. Dans un graphe multi-

étiqueté, les sommets et les arcs sont associés à des étiquettes décrivant leurs propriétés. Etant donnés L_V (resp. L_E) un ensemble d'étiquettes de sommets (resp. d'arcs), un **graphe multi-étiqueté** est défi ni par un triplet $G = \langle V, r_V, r_E \rangle$ tel que :

- V est un ensemble de sommets,
- $r_V \subseteq V \times L_V$ est une relation associant les sommets à leurs étiquettes, *i.e.,* r_V est l'ensemble des couples (v, l) tels que le sommet v a pour étiquette l,
- $r_E \subseteq V \times V \times L_E$ est une relation associant les arcs à leurs étiquettes, *i.e.,* r_E est l'ensemble des triplets (v, v', l) tels que l'arc (v, v') a pour étiquette l. Sans perte de généralité, on supposera que chaque arc possède au moins une étiquette. Par conséquent, l'ensemble E des arcs est défi ni par $E = \{(v, v')|\exists l, (v, v', l) \in r_E\}$.

Les tuples de r_V et r_E constituent les caractéristiques de G. L'ensemble $descr(G) = r_V \cup r_E$ contient toutes ces caractéristiques des sommets et des arcs de G et décrit entièrement le graphe G.

Nous introduisons maintenant la mesure de similarité de graphes de Champin & Solnon (2003) ; nous invitons le lecteur à se référer à cet article pour plus de détails. Cette mesure de similarité est défi nie pour deux graphes multi-étiquetés $G = \langle V, r_V, r_E \rangle$ et $G' = \langle V', r_{V'}, r_{E'} \rangle$ défi nis sur les mêmes ensembles L_V et L_E d'étiquettes et tels que $V \cap V' = \emptyset$.

Pour mesurer la similarité de deux graphes, une première étape consiste à apparier leurs sommets. L'appariement considéré ici est multivalent, *i.e.*, chaque sommet d'un graphe est apparié à un ensemble éventuellement vide de sommets de l'autre graphe. Plus formellement, un **appariement multivoque** de deux graphes G et G' est une relation $m \subseteq V \times V'$ contenant tous les couples $(v, v') \in V \times V'$ tels que le sommet v est apparié au sommet v'.

Etant donné un appariement m, l'étape suivante consiste à identifi er l'ensemble des caractéristiques communes aux deux graphes par rapport à m. Cet ensemble contient toutes les caractéristiques des sommets (resp. arcs) de G et de G' appariés dans m à au moins un sommet (resp. arc) ayant la même caractéristique. Plus formellement, l'ensemble $descr(G) \sqcap_m descr(G')$ des caractéristiques communes à G et G' par rapport à l'appariement m est défi ni par :

$$\begin{aligned} descr(G) \sqcap_m descr(G') \quad \doteq \quad & \{(v, l) \in r_V | \ \exists (v, v') \in m, \ (v', l) \in r_{V'}\} \\ \cup \quad & \{(v', l) \in r_{V'} | \ \exists (v, v') \in m, \ (v, l) \in r_V\} \\ \cup \quad & \{(v_i, v_j, l) \in r_E | \ \exists (v_i, v'_i) \in m, \exists (v_j, v'_j) \in m, (v'_i, v'_j, l) \in r_{E'}\} \\ \cup \quad & \{(v'_i, v'_j, l) \in r_{E'} | \ \exists (v_i, v'_i) \in m, \exists (v_j, v'_j) \in m, (v_i, v_j, l) \in r_E\} \end{aligned}$$

Etant donné un appariement multivoque m, nous devons aussi identifi er l'ensemble des sommets éclatés (les *splits*), *i.e.*, l'ensemble des sommets appariés à plus d'un sommet. Chaque sommet éclaté v est associé à l'ensemble s_v des sommets auquel il est apparié :

$$\begin{aligned} splits(m) = \quad & \{(v, s_v) \quad | \ v \in V, \quad s_v = \{v' \in V' | (v, v') \in m\}, \quad |s_v| \geq 2\} \\ \cup \quad & \{(v', s_{v'}) \quad | \ v' \in V', \quad s_{v'} = \{v \in V | (v, v') \in m\}, \quad |s_{v'}| \geq 2\} \end{aligned}$$

La **similarité** de G et G' par rapport à un appariement m est alors défi nie par :

$$sim_m(G, G') \quad = \quad \frac{f(descr(G) \sqcap_m descr(G')) - g(splits(m))}{f(descr(G) \cup descr(G'))} \tag{1}$$

où f et g sont deux fonctions dépendantes de l'application considérée. Par exemple, si f est la fonction cardinalité et g la fonction nulle, la similarité est alors proportionnelle au

nombre de caractéristiques communes par rapport au nombre total de caractéristiques. Si g est une fonction de cardinalité, alors la similarité décroîtra proportionnellement au nombre de sommets éclatés.

Finalement, la similarité $sim(G, G')$ de deux graphes G et G' est défi nie comme la plus grande similarité possible, *i.e.*, celle obtenue par le meilleur appariement :

$$sim(G, G') = \max_{m \subseteq V \times V'} \frac{f(descr(G) \sqcap_m descr(G')) - g(splits(m))}{f(descr(G) \cup descr(G'))} \quad (2)$$

3 Généricité de cette mesure

La mesure de similarité décrite en section 2 a été initialement proposée pour comparer des objets de conception. Cependant, cette mesure est générique dans le sens où elle est paramétée par les fonctions f et g. Dans cette section, nous montrons comment ces fonctions peuvent être défi nies pour instancier cette mesure générique en d'autres mesures de similarité couramment utilisées. Ces mesures sont souvent défi nies pour des graphes non-étiquetés. Nous supposerons donc qu'un graphe non-étiqueté est un cas particulier de graphe étiqueté où tous les sommets (resp. arcs) ont la même étiquette l_v (resp. l_e).

Isomorphisme de graphes. Un graphe $G = (V, E)$ est isomorphe à un graphe $G' = (V', E')$ ssi $|V| = |V'|$ et s'il existe une fonction bijective $\phi : V \to V'$ telle que $(v_1, v_2) \in E$ ssi $(\phi(v_1), \phi(v_2)) \in E'$. Autrement dit, le problème consiste à trouver un appariement univoque permettant de retrouver tous les sommets et tous les arcs des deux graphes.

Si les fonctions f et g de la formule (2) sont défi nies comme la fonction de cardinalité, alors $sim(G, G')$=1 ssi il existe un appariement m tel que $descr(G) \sqcap_m descr(G') = descr(G) \cup descr(G')$ (*i.e.*, m permet de retrouver toutes les caractéristiques de G et G') et $splits(m) = \emptyset$, (*i.e.*, m est univoque). Autrement dit, $sim(G, G') = 1$ ssi G et G' sont isomorphes.

Isomorphisme de sous-graphes partiel. Un graphe $G = (V, E)$ est un sous-graphe partiel du graphe $G' = (V', E')$ ssi $|V| \leq |V'|$ et s'il existe une fonction injective $\phi : V \to V'$ telle que $(v_1, v_2) \in E \Rightarrow (\phi(v_1), \phi(v_2)) \in E'$. Autrement dit, le problème consiste à trouver un appariement univoque des sommets de G aux sommets de G' permettant de retrouver toutes les caractéristiques de G.

Si la fonction g de la formule (2) est la fonction cardinalité et si f est une fonction ne comptant que les caractéristiques de G, *i.e.*, f est défi nie comme une somme pondérée où le poids des caractéristiques de G (resp G') est 1 (resp 0), alors $sim(G, G') = 1$ ssi il existe un appariement m tel que $descr(G) \subseteq descr(G) \sqcap_m descr(G')$ (*i.e.*, m permet de retrouver toutes les caractéristiques de G) et $splits(m) = \emptyset$ (*i.e.*, m est univoque). Autrement dit, $sim(G, G') = 1$ ssi il existe un isomorphisme de sous-graphes partiel de G dans G'.

Isomorphisme de sous-graphes. Le problème de l'isomorphisme de sous-graphes est un cas particulier du problème de l'isomorphisme de sous-graphes partiel où la condition suivante est ajoutée : tous les couples de sommets de G qui ne sont pas reliés par un arc doivent être appariés à des sommets de G' qui ne sont également pas reliés par un arc (*i.e.*, $\forall (v_1, v_2) \in V^2, (v_1, v_2) \notin E \Rightarrow (\phi(v_1), \phi(v_2)) \notin E'$).

Pour vérifi er cette condition, il faut ajouter à tous les couples de sommets non reliés par un arc une étiquette "pas-un-arc" puis s'assurer que le meilleur appariement retrouve ces étiquettes. Plus formellement, étant donné un graphe $G = (V, E)$, nous défi nissons le graphe étiqueté $G_{label} = (V, r_V, r_E)$ tel que $r_V = \{(v, l_v)|v \in V\}$ et $r_E = \{(u, v, l_e)|(u, v) \in E\} \cup \{(u, v, l_{notE})|(u, v) \in V \times V - E\}$. Si les fonctions f et g sont défi nies comme pour le problème de l'isomorphisme de sous-graphes partiel, $sim(G_{label}, G'_{label}) = 1$ ssi il existe un isomorphisme de sous-graphes de G dans G'.

Plus grand sous-graphe partiel commun ($mcps$). Le $mcps$ de deux graphes G et G' est le plus grand graphe (par rapport au nombre de sommets et d'arcs) qui soit isomorphe à des sous-graphes partiels de G et de G'.

Défi nissons la fonction f de la formule (2) comme une fonction de cardinalité et la fonction g telle que tout éclatement de sommet soit interdit (*i.e.*, $g(S) = +\infty$ si $S \neq \emptyset$ et $g(\emptyset) = 0$). L'appariement m qui maximise la formule (1) est alors l'appariement permettant de retrouver un maximum de caractéristiques de sommets et d'arcs tout en interdisant les éclatements de sommets. Cet appariement m correspond donc à un $mcps$.

Plus grand sous-graphe commun (mcs). En s'inspirant du problème de l'isomorphisme de sous-graphes non partiel, il est possible de résoudre le problème de la recherche du plus grand sous-graphe non partiel commun (mcs) à deux graphes. Pour cela, il faut défi nir la fonction g de telle façon que les éclatements de sommets soient interdits (*i.e.*, $g(S) = +\infty$ si $S \neq \emptyset$ et $g(\emptyset) = 0$). La fonction f doit "vérifi er" que seuls des couples de sommets de même types (*i.e.*, reliés ou pas à un arc) sont appariés entre eux. Autrement dit, la fonction f doit retourner une valeur nulle s'il existe un couple de sommets (u, v) de G (resp. de G') apparié à un couple de sommet (u', v') de G' (resp. de G) tel que la relation entre u et v (relié ou pas par un arc) n'est pas la même que la relation entre u' et v'. Dans tous les autres cas, f doit retourner une valeur proportionnelle au nombre de sommets appariés. De façon plus formelle, en utilisant les graphes G_{label} et G'_{label} défi nis pour l'isomorphisme de sous-graphes non partiel, $f(S) = 0$ si $\exists(u, v) \in V_1^2 \cup V_2^2$ tel que $\{(u, l_v), (v, l_v)\} \subseteq S$ (*i.e.*, les sommets u et v sont appariés) et que $\{(u, v, l_e), (u, v, l_{notE})\} \cap S = \emptyset$ (*i.e.*, aucune propriété reliant u à v n'a été retrouvée). Dans tous les autres cas, $f(S) = |\{u|(u, l_v) \in S\}|$. Avec de telles fonctions f et g, le meilleur appariement m correspond au mcs.

Distance d'édition de graphes (ged). La distance d'édition (ged) entre deux graphes G et G' est le coût minimal pour transformer G en G'. Pour cette transformation, on dipose de six opérations élémentaires : l'insertion, la suppression et le réétiquetage de sommets et d'arcs. Chaque opération a un coût. L'ensemble d'opérations le moins couteux permettant de transformer G en un graphe isomorphe à G' défi nit la distance d'édition entre G et G'. Bunke (1997) montre que la distance d'édition est un concept très proche de la mesure de similarité basée sur le plus grand sous-graphe partiel commun à deux graphes : elle est donc également très proche de la mesure de similarité de la formule (2).

Si les graphes ne sont pas étiquetés, seules les opérations de suppression et d'insertion sont utilisées. Nous pouvons remarquer que, étant donné un appariement m, l'ensemble des caractéristiques de sommets et d'arcs contenues dans $descr(G) - (descr(G) \sqcap_m descr(G'))$ (resp. dans $descr(G') - (descr(G) \sqcap_m descr(G'))$) correspond à l'ensemble des opérations de suppression (resp. d'insertion) d'arcs et de sommets néces-

saires pour rendre le graphe G isomorphe au graphe G'. Choisissons g de telle façon que les éclatements de sommets soient interdits et la fonction f de la formule (2) comme une fonction de pondération où les poids sont défi nis en fonction des coûts des transformations élémentaires. L'appariement m qui maximise la formule (1) défi nit l'ensemble des opérations élémentaires qui minimise la distance d'édition.

Si nous considérons maintenant la distance d'édition appliquée à des graphes mono-étiquetés (*i.e.*, des graphes où chaque sommet et chaque arc possède une seule étiquette), les opérations de substitution (moins coûteuses qu'une suppression suivie d'une insertion) peuvent s'appliquer. Cependant, lorsque les sommets (resp. les arcs) sont mono-étiquetés, les informations contenues dans l'ensemble $descr(G) \sqcap_m descr(G')$ des caractéristiques communes ne permettent pas de différencier le cas où deux sommets (resp. arcs) n'ayant pas la même étiquette sont appariés du cas où ces deux sommets (resp. arcs) ne sont pas appariés (dans les deux cas, aucune des deux étiquettes n'est retrouvée). Afi n de détecter les substitutions, il est nécessaire d'ajouter à tous les sommets (resp. tous les arcs) une étiquette commune l_v (resp. l_e) en plus de leur étiquette d'origine. Ces étiquettes permettent alors de savoir si un sommet (resp. un arc) est apparié ou non. Si l'étiquette l_v d'un sommet u (resp. l'étiquette l_e d'un arc (u, v)) n'appartient pas à $descr(G) \sqcap_m descr(G')$, alors le sommet u (resp. l'arc (u, v)) a été supprimé ou inséré. A contrario, lorsque cette étiquette l_v (resp. l_e) est retrouvée, soit il y a eu substitution – et l'étiquette d'origine de u (resp. (u, v)) n'est pas retrouvée – soit il n'y a pas eu d'opération de transformation sur u (resp. (u, v)). La fonction de pondération f peut être choisie de telle sorte que les poids reproduisent fi dèlement les coûts de tous les types de transformation élémentaire.

Distance d'edition étendue. Afi n de comparer des images sur- et sous- segmentées, Ambauen *et al.* (2003) propose une distance d'édition de graphe étendue autorisant deux nouvelles opérations par rapport à la distance d'édition classique : l'éclatement de sommets –reliant un sommet de G à plusieurs sommets de G'– et la fusion de sommets –reliant plusieurs sommets de G à un même sommet de G'.

Etant donné un appariement m, l'ensemble des couples $(v, s_v) \in splits(m)$ tels que $v \in G$ correspond aux opérations d'éclatement de sommets et l'ensemble des couples $(v', s_{v'})$ tels que $v' \in G'$ correspond aux fusions de sommets. Dés lors, si la fonction g de la formule (2) est défi nie comme une fonction de pondération où les poids correspondent aux coûts d'éclatement et de fusion de sommets et si la fonction f et les graphes sont défi nis comme pour la distance d'édition non étendue, alors l'appariement qui maximise la formule (1) permet de calculer la distance d'édition étendue telle que défi nie dans Ambauen *et al.* (2003).

Appariement non bijectif. Ce problème est introduit dans Boeres *et al.* (2004) afi n de trouver le meilleur appariement entre les régions d'une image schématique d'un cerveau et les régions d'une image réelle du cerveau. L'image réelle est bruitée (contrairement à l'image schématique) et donc sur-segmentée si bien qu'une région du schéma correspond souvent à plusieurs régions de l'image. Etant donné un graphe modèle $G = (V, E)$ et un graphe image $G' = (V', E')$, les appariements autorisés sont défi nis par une fonction $\phi : V \to \wp(V')$ associant à chaque sommet de V un ensemble non vide de sommets de V' et telle que (i) chaque sommet de V' est associé à exactement un sommet de V, (ii) quelques couples de sommets (v, v') jugés trop différents sont interdits (*i.e.*, $v' \notin \phi(v)$)

et (iii), le sous-graphe de G' induit par chaque ensemble $\phi(v)$ doit être connexe (afi n de ne fusionner que des régions adjacentes). Un poids (éventuellement négatif) $s^v(v_i, v'_i)$ (resp. $s^e(e_i, e'_i)$) est associé à chaque couple de sommets $(v_i, v'_i) \in V \times V'$ (resp. à chaque couple d'arcs $(e_i, e'_i) \in E \times E'$). L'objectif est de trouver un appariement respectant les contraintes et maximisant la somme des poids des couples de sommets et d'arcs appariés.

Les fonctions f et g peuvent être défi nies de telle façon que l'appariement m qui maximise la fonction (2) corresponde au meilleur appariement au sens de Boeres *et al.* (2004). Afi n de prendre en compte le fait que des poids sont associés aux couples de sommets et d'arcs et que la condition (ii) est respectée, il convient d'ajouter des étiquettes aux sommets (resp. aux arcs) de telle façon qu'une étiquette $l_{(v,v')}$ (resp. $l_{(e,e')}$) appartienne à l'ensemble $descr(G) \sqcap_m descr(G')$ des caractéristiques communes ssi v est apparié à v' (resp. e à e'). Pour cela, il suffi t d'ajouter à tous les sommets $v \in V$ (resp. $v' \in V'$) des étiquettes $l_{(v,v')}$ telles que $v' \in V'$ (resp. $v \in V$) et à tous les arcs $e \in E$ (resp. $e' \in E'$) des étiquettes $l_{(e,e')}$ telles que $e' \in E'$ (resp. $e \in E$). Ainsi, l'étiquette $l_{(v,v')}$ d'un sommet v (resp. l'étiquette $l_{(e,e')}$ d'un arc e) est contenue dans $descr(G) \sqcap_m descr(G')$ ssi le sommet v est apparié au sommet v' (resp. l'arc e est apparié à l'arc e'). La fonction f peut donc déduire l'appariement réalisé de l'ensemble $descr(G) \sqcap_m descr(G')$ et calculer alors la même mesure que Boeres *et al.* (2004). La fonction f est défi nie comme une fonction de pondération associant des poids aux étiquettes $l_{(v,v')}$ (resp. $l_{(e,e')}$) en fonction de $s^v(v, v')$ (resp. $s^e(e, e')$) ou un poids très fortement négatif s'il est interdit d'apparier v à v' ou s'il existe un sommet non apparié (contrainte (i)). La fonction g permet de vérifi er les contraintes (i) et (iii). Lorsqu'un appariement m associe un sommet v de l'image à plus d'un sommet (la contrainte (i) est alors violée), il existera un couple (v, s_v) dans l'ensemble $splits(m)$ auquel la fonction g pourra donner un poids infi ni (interdisant ainsi l'appariement m). Lorsqu'un sommet v du modèle est apparié à un ensemble s_v de sommets de l'image, le couple (v, s_v) est contenu dans $splits(m)$. La fonction g peut alors vérifi er que le sous-graphe de l'image induit par les sommets de s_v est connexe et vérifi er ainsi que la contrainte (iii) n'est pas violée. Notons enfi n que comme le meilleur appariement au sens de notre mesure de similarité correspond au meilleur appariement au sens du problème de l'appariement non-bijectif de graphes, il est possible de calculer la mesure de Boeres *et al.* (2004) à partir de notre mesure de similarité.

Discussion

Les mesures de similarité de graphes proposées dans Ambauen *et al.* (2003) et Boeres *et al.* (2004) sont basées sur des appariement multivoques (*i.e.*, un sommet peut être apparié à plusieurs autres). Ces deux mesures ont été introduites en reconnaissance d'images pour la prise en compte des problèmes de sur-segmentation des images : les appariements multivoques permettent d'apparier une région d'une image sous-segmentée à plusieurs régions d'une image sur-segmentée.

Ces mesures sont spécifi ques aux problèmes pour lesquelles elles ont été défi nies. Par exemple, Boeres *et al.* (2004) cherche à comparer une image réelle avec sa représentation schématique. Dans ce contexte, une région de l'image réelle doit être appariée à

une et une seule région du schéma alors qu'une région du schéma peut correspondre à plusieurs régions de l'image réelle. La mesure de similarité proposée et les algorithmes de résolution ont été construits autour de ces contraintes spécifi ques et sont diffi cilement adaptables à un autre contexte.

A contrario, la mesure de similarité proposée dans Champin & Solnon (2003) est générique. Cette mesure de similarité est paramétrée par deux fonctions f et g qui permettent d'exprimer les contraintes et les préférences propres à un problème donné (*e.g.* la recherche d'un appariement univoque ou multivoque, l'évaluation de la qualité d'un appariement...) sont exprimées par l'intermédiaire des deux fonctions f et g. Un algorithme permettant de calculer cette similarité peut donc être utilisé dans de nombreuses applications sans effort d'adaptation. En contrepartie de cette généricité, cet algorithme sera sans doute moins effi cace que des algorithmes dédiés capables d'exploiter les connaissances dépendantes du domaine d'application pour accélérer la recherche du meilleur appariement.

4 Algorithmes de mesure de similarité de graphes

Tous les problèmes d'appariement présentés en section 3 sont NP-complets ou NP-diffi ciles à l'exception du problème de l'isomorphisme de graphes dont la complexité n'est pas clairement établie. Pour certains graphes (tels que les arbres ou les graphes planaires) certains de ces problèmes deviennent polynomiaux (Aho *et al.* (1974); Hopcroft & Wong (1974); Luks (1982)).

Les problèmes d'isomorphismes de graphes et de sous-graphes sont généralement aisément résolus par des algorithmes complets. Des techniques très effi caces d'étiquetages des sommets ont été proposés par McKay (1981) et Sorlin & Solnon (2004) propose d'utiliser la programmation par contraintes et une méthode de fi ltrage ad-hoc pour le problème de l'isomorphisme de graphes. Des algorithmes complets ont été proposés pour la recherche de l'appariement qui maximise la formule (1) (Champin & Solnon (2003)) et pour le calcul de la distance étendue entre deux graphes (Ambauen *et al.* (2003)). Ces algorithmes sont basés sur une exploration exhaustive de l'espace de recherche combinée à des techniques de fi ltrage. Ils garantissent l'optimalité de la solution trouvée mais, du fait de l'explosion combinatoire qu'ils engendrent, ils sont limités à de très petits graphes. L'utilisation d'algorithmes incomplets qui ne garantissent pas l'optimalité de la solution trouvée mais ayant une complexité polynomiale semble être une bonne alternative. Par exemple, Boeres *et al.* (2004) propose un algorithme de construction aléatoire d'appariements non-bijectifs ainsi qu'un algorithme de recherche locale améliorant ces appariements jusqu'à l'obtention d'un maximum local. Ces deux algorithmes sont dédiés à l'appariement d'une image réelle à son modèle.

Dans cette section, nous décrivons trois algorithmes incomplets –un algorithme glouton, une recherche taboue et une recherche taboue réactive– pour le calcul de la mesure de similarité de deux graphes étiquetés. Ces algorithmes sont génériques dans le sens où ils sont paramétrés par les fonctions f et g utilisées pour introduire les connaissances et les contraintes dépendantes à une application. Ils peuvent donc être utilisés pour résoudre tout type de problèmes d'appariement de graphes.

Algorithme glouton

Cet algorithme a été proposé dans Champin & Solnon (2003). Nous le présentons brièvement car il est utilisé comme un point de départ de nos algorithmes de recherche taboue. Pour plus de précisions, nous renvoyons le lecteur à l'article original.

L'algorithme démarre d'un appariement vide $m = \emptyset$. A chaque itération, il ajoute à m un couple de sommets parmi l'ensemble $cand = V \times V' - m$ des candidats. Le couple à ajouter est choisi selon une heuristique gloutonne : le sous-ensemble des candidats dont l'ajout maximise la similarité (*i.e.*, la formule (1)) est tout d'abord construit. Ce sous-ensemble contient généralement plus d'un candidat. Afi n de les départager, le "potentiel" de chaque candidat (v, v') est anticipé en prenant en compte les caractéristiques d'arcs entrants (resp. sortants) communs à v et v' et n'étant pas encore dans $descr(G) \sqcap_{m \cup \{(v,v')\}} descr(G')$. En cas d'ex-æquo, le couple de sommets à ajouter est choisi aléatoirement. Ces ajouts gloutons sont répétés jusqu'à l'obtention d'un maximum local, *i.e.*, jusqu'à ce qu'aucun couple de sommets ne puisse augmenter la similarité. Cet algorithme a une complexité en temps polynomiale de $\mathcal{O}((|V| \times |V'|)^2)$ (lorsque le calcul de f et de g est de complexité linéaire en temps). En contrepartie de cette faible complexité, l'algorithme ne fait jamais de "retour arrière" et n'est pas complet : bien qu'il puisse parfois trouver la meilleure solution, cela n'est pas systématique et on ne peut pas l'utiliser pour prouver l'éventuelle optimalité d'une solution. Cet algorithme n'étant pas déterministe, nous pouvons l'exécuter plusieurs fois et garder la meilleure solution trouvée.

Recherche locale

L'algorithme glouton retourne un appariement "localement optimal" dans le sens où il ne peut pas être amélioré en ajoutant un seul couple de sommets. Il est néanmoins possible de l'améliorer en ajoutant et en supprimant plusieurs couples de sommets. Une recherche locale tente d'améliorer une solution en explorant son voisinage. Les voisins d'un appariement m sont les appariements qui peuvent être obtenus en ajoutant ou en enlevant un couple de sommets à m.

$$\begin{aligned} \forall m \in \wp(V \times V'), voisinage(m) &= \{m \cup \{(v,v')\} | (v,v') \in (V \times V') - m\} \\ &\cup \{m - \{(v,v')\} | (v,v') \in m\} \end{aligned}$$

A partir d'un appariement initial calculé par l'algorithme glouton, l'espace de recherche est exploré de voisin en voisin jusqu'à ce que la meilleure solution soit obtenue (lorsque la qualité de celle-ci est connue) ou jusqu'à ce que le nombre d'itérations maximum soit atteint. A chaque itération, une heuristique guide la recherche en déterminant le prochain voisin à explorer.

Méta-heuristique taboue

La recherche *taboue* (Glover (1989); Dorne & Hao (1998); Petrovic *et al.* (2002)) est une des meilleures heuristiques connues pour choisir le prochain voisin à explorer.

A chaque itération, le voisin est choisi par rapport au critère de l'algorithme glouton. Notons cependant que le meilleur voisin d'un appariement m localement optimal est de moins bonne qualité que m. Afi n de ne pas cantonner la recherche autour d'un maximum local en ajoutant puis en retirant continuellement le même couple de sommets, une liste taboue est utilisée. Cette liste de longueur k mémorise les k derniers mouvements effectués (*i.e.*, les k derniers couples de sommets ajoutés ou supprimés) afi n d'interdire un mouvement inverse à un mouvement récemment effectué (*i.e.*, ajouter/supprimer un couple de sommets récemment supprimé/ajouté). Une exception nommée "aspiration" est ajoutée : si un mouvement interdit permet d'atteindre un appariement de meilleur qualité que le meilleur appariement connu jusqu'alors, le mouvement est quand même réalisé. La fi gure 1 décrit l'algorithme tabou du calcul d'une valeur approchée de la formule (2).

fonction Tabou($G = \langle V, r_V, r_E \rangle, G' = \langle V', r_{V'}, r_{E'} \rangle, k, limiteQualite, maxMouv$)
retourne un appariement $m \subseteq V \times V'$
  $m \leftarrow Glouton(G, G')$; $best_m \leftarrow m$; $nbMouv \leftarrow 0$
  tant que $sim_{best_m}(G, G') < limiteQualite$ **et** $nbMouv < maxMouv$ **faire**
    $cand \leftarrow \{m' \in voisinage(m) / sim_{m'}(G, G') > sim_{best_m}(G, G')\}$
    si $cand = \emptyset$ **alors**/* *pas d'aspiration* */
      $cand \leftarrow \{m' \in voisinage(m) / pasTabou(m, m', k)\}$
    fin si
    $cand \leftarrow \{m' \in cand / m'$ est maximal par rapport au critère de glouton$\}$
    choisir aléatoirement $m' \in cand$
    $rendreTabou(m, m', k)$; $m \leftarrow m'$; $nbMouv \leftarrow nbMouv + 1$
    si $sim_m(G, G') > sim_{best_m}(G, G')$ **alors** $best_m \leftarrow m$ **fin si**
  fin tant que
  retourner $best_m$

FIG. 1 – Algorithme Tabou

Recherche taboue réactive

La longueur k de la liste taboue est un paramètre critique et diffi cile à déterminer : si la liste est trop longue, la diversifi cation de la recherche est trop forte et l'algorithme converge trop lentement ; si la liste est trop courte, l'intensifi cation de la recherche est trop forte et l'algorithme reste autour d'un optimum local et ne trouve plus de meilleures solutions. Battiti & Protasi (2001) résolvent ce problème grâce à la recherche taboue réactive dans laquelle la longueur de la liste taboue est adaptée dynamiquement pendant la recherche. Si un même appariement est exploré plusieurs fois, la recherche doit être diversifi ée. Afi n de détecter une telle redondance, la clé de hachage de chaque appariement visité est mémorisée. Quand une collision a lieu dans la table de hachage, la liste taboue est allongée afi n de diversifi er la recherche. A contrario, quand il n'y a pas eu de collisions durant un certain nombre de mouvements (signe que la recherche est suffi samment diversifi ée) la liste taboue est raccourcie. Les clés de hachage pouvant être calculées de façon incrémentale, le coût supplémentaire de ces calculs est négligeable.

5 Résultats expérimentaux

5.1 Problèmes étudiés

Nous avons expérimenté les algorithmes sur 3 types d'appariements différents :

1. Des problèmes d'isomorphisme de graphes et de sous-graphes proposés par Foggia *et al.* (2001). Les graphes ont entre 100 et 200 sommets pour les problèmes d'isomorphisme de graphes. Les problèmes d'isomorphisme de sous-graphes recherchent un sous-graphe d'un graphe ayant entre 20 et 100 sommets. Les sous-graphes ont 80%, 60% ou 40% de sommets en moins que les graphes qui les contiennent.
2. Sept instances du problème d'appariement non-bijectif de graphes proposées par Boeres *et al.* (2004). Le graphe schéma a entre 10 et 50 sommets, le graphe image a entre 30 et 250 sommets. Pour plus de détails, se référer à Boeres *et al.* (2004).
3. Une centaine d'instances du problème d'appariement multivoque de graphes. Il n'existe pas de benchmark de problèmes d'appariements multivoques. Nous avons donc conçu un générateur de paires de graphes "similaires". Un graphe est généré aléatoirement puis quelques fusions et éclatements de sommets et quelques insertions et suppressions de sommets lui sont appliqués afi n d'obtenir un second graphe similaire au premier. Une borne mimimum de la similarité des deux graphes est calculée à partir de l'ensemble des transformations effectuées. Le problème est considéré résolu lorsque cette borne est atteinte. Quand les composants des graphes sont très différents de par leurs étiquettes, le meilleur appariement est facilement trouvé car le nombre de couples de sommets qu'il peut être intéressant d'apparier est faible. Afi n d'obtenir des instances diffi ciles les graphes générés sont tels que tous les arcs et tous les sommets ont la même étiquette. Les graphes considérés ici ont 100 sommets et entre 200 et 360 arcs. Le second graphe est obtenu en fusionnant ou éclatant 5 sommets et en supprimant ou ajoutant 10 arcs ou sommets. Les fonctions f et g de la formule (2) sont des fonctions de cardinalité.

5.2 Protocole expérimental

Les trois algorithmes ont été écrits en C++ et exécutés sur un Pentium IV 2GHz avec 512Mo de RAM. Les problèmes sont modélisés en problèmes de mesure de similarité et sont tous résolus par le même programme. Une seule modifi cation du code (facultative) a été réalisée pour le problème de Boeres *et al.* (2004). Cette modifi cation permet d'abstraire les étiquettes afi n d'accélérer le calcul de la similarité par rapport à un appariement.

Une résolution par l'algorithme glouton consiste à exécuter 500 fois l'algorithme et à retourner le meilleur appariement trouvé. Les recherches locales exécutent le même nombre de mouvements que l'algorithme glouton, *i.e.*, $500 * n$ si les appariements trouvés par l'algorithme glouton ont en moyenne n couples de sommets. En moyenne, les meilleurs résultats de la version non réactive de tabou ont été obtenus avec une liste taboue de longueur $k = 30$. Cette valeur n'étant cependant pas optimale pour toutes

les instances, les tests ont aussi été effectués pour des valeurs de k comprises entre 10 et 50. Les meilleurs paramètres trouvés pour la liste taboue réactive sont 10 (resp. 50) pour la longueur minimale *lMin* (resp. maximale *lMax*), 15 pour la longueur *lDiff* d'allongement ou de raccourcissement et 1000 mouvements pour la fréquence *freq* de raccourcissement de la liste.

5.3 Résultats

Isomorphismes. En 10 secondes, l'algorithme glouton résoud 80% des instances d'isomorphisme de graphes alors que les recherches taboues (réactives ou non) en résolvent 100%. Les problèmes d'isomorphisme de sous-graphes sont beaucoup plus difficiles : en 200s, la recherche taboue réactive résoud 66% des instances à 100 sommets alors que glouton n'en résoud que 4,4%. Ces résultats mitigés s'expliquent par le fait que notre algorithme n'utilise aucune technique de filtrage et explore donc potentiellement tous types d'appariements, même ceux qui sont multivoques.

Notons qu'une variation (même légère) du paramètre k de tabou non-réactif influence énormément les résultats et que la valeur optimale de k varie d'une instance à l'autre d'un même problème. A contrario, les paramètres de tabou réactif sont plus "robustes" : de petites variations de ces paramètres ne changent pas significativement les résultats.

Appariement non-bijectif de graphes. Le tableau de la figure 2 résume les résultats obtenus sur le problème de l'appariement non-bijectif de graphes. La colonne *Pb* donne le numéro de l'instance. La colonne *fLS+* donne la similarité maximum obtenue par Boeres *et al.* (2004), la colonne *T* (resp. *TR*) donne le résultat de tabou non réactif (resp. réactif) dans son paramétrage "standard" (voir la section 5.2). La colonne *T** donne le résultat et la longeur de la liste du meilleur tabou non réactif. La colonne *TR** donne le résultat et le paramétrage (sous la forme *lMin.lMax.lDiff.freq*) du meilleur tabou réactif. Notons que l'exécution de nos trois algorithmes sur ces instances est déterministe : les poids manipulés par ces instances sont des réels et il n'y a jamais eu d'ex-æquo. Aucun couple n'est donc choisi aléatoirement lors de l'exécution de ces instances.

Pb	fLS+	TR	TR*	T	T*
5	.5474	.5481	.5548(10.50.15.750)	.5463	.5463(30)
5a	.5435	.5529	.5597(10.50.15.750)	.5519	.5558(20)
6	.4248	.4213	.4213(Tous)	.4213	.4213(Tous)
7	.6319	.6333	.6354(10.50.15.500)	.6342	.6344(35)
8	.5186	.5210	.5212(10.50.10.500)	.5195	.5204(45)
8a	.5222	.5245	.5248(10.50.10.1000)	.5231	.5240(50)
9	.5187	.5199	.5202(15.100.20.750)	.5198	.5198(50)

FIG. 2 – Problème d'appariements non-bijectifs

Pour 5 instances sur 7, tabou non réactif obtient de meilleurs résultats que la recherche locale de Boeres *et al.* (2004). Toute instance confondue, les meilleurs résultats *en moyenne* sont obtenus avec une longueur de liste k à 30. Cependant, les informations de la colonne *T** montre que ce paramétrage ne permet d'obtenir le meilleur résultat par instance que pour 2 instances sur 7 : la longueur optimale de la liste taboue varie entre 20 et 50 selon les instances. Un paramétrage standard de tabou réactif permet d'obtenir

de meilleurs résultats que Boeres *et al.* (2004) et que tabou non réactif pour 6 instances sur 7. Ces résultats peuvent encore être améliorés avec un paramétrage fi n (colonne *TR**). Seule l'instance 6 pose problème à nos algorithmes (aucune de nos recherches locales n'a réussi à améliorer l'appariement proposé par glouton). La généricité et l'effi cacité de notre approche a un prix : les temps d'exécution de nos recherches locales sont souvent supérieurs à ceux de Boeres *et al.* (2004). Bien que la comparaison des temps d'exécution soit diffi cile (les machines utilisées n'étant pas les mêmes), le temps d'exécution moyen de tabou réactif est de 20s environ contre 11,7s pour l'algorithme de Boeres *et al.* (2004).

Appariements multivoques. Chaque algorithme a été exécuté 200 fois sur chacune des 100 instances générées. 51% des intances se sont avérées "faciles" dans le sens où elles sont toujours résolues par l'algorithme glouton. Sur les 49 instances restantes, 35 ont été facilement –et tout le temps– résolues par chacune des recherches locales (moins de 500 mouvements exécutés en moins de 4s). Les 14 dernières instances se sont avérées beaucoup plus diffi ciles et nécessitent plus de 25000 mouvements pour être résolues. La version non réactive de tabou réussit dans 64% des exécutions alors que la version réactive obtient un succès dans 79% des exécutions. En outre, la version réactive de tabou semble plus robuste que la version non réactive : le choix des paramètres a une moindre influence sur les résultats. La version réactive est donc plus effi cace et plus facilement paramétrable que la version non réactive.

6 Conclusion

Nous avons montré que la mesure de similarité de Champin & Solnon (2003) est plus générique que les autres mesures de similarité de graphes existantes (Bunke (2000); Conte *et al.* (2004)). Cette mesure est basée sur des appariements multivoques des sommets des graphes ce qui permet de prendre en compte les problèmes de granularité (en représentation de connaissances) ou les problèmes de sur- et sous-segmentation des images en reconnaisssance d'images (Ambauen *et al.* (2003); Boeres *et al.* (2004)). Nous présentons trois algorithmes de complexité polynomiale : un algorithme glouton, une recherche locale basée sur la méta-heuristique taboue et une version améliorée de cette dernière nommée "recherche taboue réactive". Nous montrons l'effi cacité de notre approche sur trois types de problèmes et en particulier sur le problème proposé par Boeres *et al.* (2004).

La mesure de similarité de Champin & Solnon (2003) est défi nie pour des graphes multi-étiquetés : à chaque sommet et chaque arc est associé un ensemble d'étiquettes décrivant leurs propriétés. Le multi-étiquetage des graphes permet une description très fi ne des objets. Par exemple, dans un contexte de représentation d'images segmentées, le sommet d'un graphe peut être associé à une étiquette décrivant la couleur de la région de l'image correspondante, une autre étiquette décrivant sa taille, une autre décrivant sa forme... Lors de l'appariement de deux images, il est alors possible de déterminer d'une part à quel point les images sont similaires (en comptant le nombre d'étiquettes retrouvées) et d'autre part en quoi elles sont similaires (en regardant précisement quelles étiquettes ont été retrouvées).

Notre mesure générique, associée au pouvoir d'expression des graphes multi-étiquetés

pourrait être utilisée pour défi nir une nouvelle mesure de similarité d'images. Nous envisageons donc maintenant de nous associer à des spécialistes de la reconnaissance d'image afi n de proposer de nouvelles méthodes de recherche et de classifi cation d'images.

Références

AHO A., HOPCROFT J. & ULLMAN J. (1974). *The design and analysis of computer algorithms*. Addison Wesley.

AMBAUEN R., FISCHER S. & BUNKE H. (2003). Graph Edit Distance with Node Splitting and Merging, and Its Application to Diatom Identifi cation. In *IAPR-TC15 Wksp on Graph-based Representation in Pattern Recognition*, p. 95–106.

BATTITI R. & PROTASI M. (2001). Reactive local search for the maximum clique problem. In SPRINGER-VERLAG, Ed., *Algorithmica*, volume 29, p. 610–637.

BOERES M., RIBEIRO C. & BLOCH I. (2004). A randomized heuristic for scene recognition by graph matching. In *WEA 2004*, p. 100–113.

BUNKE H. (1997). On a relation between graph edit distance and maximum common subgraph. *PRL : Pattern Recognition Letters*, **18**.

BUNKE H. (2000). Graph matching : Theoretical foundations, algorithms, and applications. In *Proc. Vision Interface 2000, Montreal*, p. 82–88.

CHAMPIN P.-A. & SOLNON C. (2003). Measuring the similarity of labeled graphs. In *5th International Conference on Case-Based Reasoning (ICCBR 2003)*, volume Lecture Notes in Artifi cial Intelligence 2689-Springer-Verlag, p. 80–95.

CONTE D., FOGGIA P., SANSONE C. & VENTO M. (2004). Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, **18**(3), 265–298.

DORNE R. & HAO J. (1998). *Tabu Search for graph coloring, T-coloring and Set T-colorings*, In *Metaheuristics 98 : Theory and Applications*, chapter 3. I.H. Osman et al. (Eds.), Kluver Academic Publishers.

FOGGIA P., SANSONE C. & VENTO M. (2001). A database of graphs for isomorphism and sub-graph isomorphism benchmarking. *3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition*, p. 176 –187.

GLOVER F. (1989). Tabu search - part I. *Journal on Computing*, p. 190–260.

HOPCROFT J. & WONG J. (1974). Linear time algorithm for isomorphism of planar graphs. 6^{th} *Annu. ACM Symp. theory of Comput.*, p. 172–184.

LUKS E. (1982). Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer System Science*, p. 42–65.

MCKAY B. (1981). Practical graph isomorphism. *Congressus Numerantium*, **30**, 45–87.

PETROVIC S., KENDALL G. & YANG Y. (2002). A Tabu Search Approach for Graph-Structured Case Retrieval. In *STAIRS 2002*, p. 55–64.

SORLIN S. & SOLNON C. (2004). A global constraint for graph isomorphism problems. In *the 6th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems (CP-AI-OR 2004)*.