



Comparaison de contenus d'images par appariement multivoque de graphes

Kieu-Diem Ho, Jean-Yves Ramel, Nicolas Monmarché

► To cite this version:

Kieu-Diem Ho, Jean-Yves Ramel, Nicolas Monmarché. Comparaison de contenus d'images par appariement multivoque de graphes. ORASIS 2021, Centre National de la Recherche Scientifique [CNRS], Sep 2021, Saint Ferréol, France. hal-03339629

HAL Id: hal-03339629

<https://hal.science/hal-03339629>

Submitted on 9 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Comparaison de contenus d'images par appariement multivoque de graphes

KD. Ho¹

JY. Ramel¹

N. Monmarché¹

¹ Laboratoire d'Informatique Fondamentale et Appliqué de Tours

Polytech Tours, Université de Tours, 37000 Tours

{ kieu.ho, jean-yves.ramel, nicolas.monmarche } @univ-tours.fr

Résumé

L'usage de graphes pour représenter le contenu sémantique des images est de plus en plus fréquent du fait du fort pouvoir de représentation et d'interprétabilité obtenues. Localiser, reconnaître ou interpréter un contenu revient alors à comparer ou mettre en correspondance des graphes. Les méthodes classiques d'appariement de graphes (dites univoques) sont peu tolérantes au bruit et ne permettent pas une bonne gestion de la sur-segmentation par exemple. Cet article présente un algorithme d'appariement multivoque de graphes résolu à l'aide d'une méthode d'optimisation par colonies de fourmis artificielles (max-min ant system). La tolérance au bruit introduite par l'autorisation d'appariements multivoque rend le problème très combinatoire puisque un nœud (arc) dans un graphe peut alors correspondre à plus d'un nœud (arc) dans l'autre graphe. Nous formalisons ce problème comme un problème de distance d'édition de graphe étendue en ajoutant des possibilités d'opérations de division et de fusion. La résolution par colonies de fourmis est combinée à une stratégie de recherche locale par voisinage pour améliorer la qualité de la solution et pour accélérer le temps de calcul. L'efficacité de l'approche proposée est vérifiée sur un ensemble d'images de symboles pré-segmentés dans des images de documents graphiques. Ces résultats montrent l'intérêt de l'utilisation d'un algorithme d'appariement multivoque pour reconnaître des formes bruitées fournissant des informations les différents éléments constituant le contenu de l'image plutôt qu'uniquement une mesure de similarité entre formes.

Mots Clef

reconnaissance de contenu d'images, appariement multivoque de graphes, distance d'édition, colonies de fourmis, documents graphiques, classification.

Abstract

The use of graphs to represent the semantic content of images is becoming more and more frequent due to the strong power of representation and interpretability obtained. Locating, recognizing or interpreting a content then comes down to comparing or matching graphs. Classical graph matching methods (called univocal) are not very tolerant to noise and do not allow a good management of

over-segmentation for example. This paper presents a multivalent graph matching algorithm using a max-min ant system optimization method. The noise tolerance introduced by allowing multivalent matching makes the problem very combinatorial since a node (edge) in one graph can then be matched to more than one node (edge) in the second graph. We formalize this problem as an extended graph edit distance problem by adding possibilities of split and merge operations. The ant colony optimization is combined with a local neighborhood search strategy to improve the quality of the solution and to speed up the computational time. The efficiency of the proposed approach is verified on a set of pre-segmented symbols in graphical document images. These results show the interest of using a multivalent matching algorithm to recognize noisy shapes providing information about the different elements constituting the content of the image rather than only a similarity measure between shapes.

Keywords

content recognition, graph matching, graph edit distance, max-min ant system, symbol recognition, document images, classification.

1 Introduction

Plusieurs applications dans la vie réelle exigent la détermination d'une mesure de similarité explicite entre les objets plutôt qu'une valeur numérique [15]. Parmi les approches existantes, l'approche basée sur les graphes est prometteuse car elle fournit la correspondance des sous-parties ainsi que la mesure de similarité. Cette méthode nécessite une première étape de construction d'une représentation pour transformer les objets ou images à comparer en graphes. Ensuite, une méthode d'appariement de graphes doit être appliquée pour mettre en relation, chercher des correspondances entre les sous-parties des graphes selon des critères visuels et topologiques. Ce type de techniques a déjà été appliqué avec succès dans certains domaines tels que l'analyse d'images 2D et 3D, les applications biologiques et biomédicales [3], [4], [15].

Un problème d'appariement de graphes (AG) peut appartenir à différentes catégories en fonction des contraintes imposées sur les mises en correspondance (c'est à dire la tolérance aux distorsions). Les méthodes d'AG exacts

n'autorisent la mise en correspondance entre sommets et arêtes de 2 graphes distincts que si, à la fois les attributs (étiquettes) sur les sommets et arêtes, et la structure du graphe sont respectés [8]. Par conséquent, elles sont bien adaptées lorsque les éléments des graphes possèdent des attributs symboliques et dans un contexte non bruité. Les méthodes d'AG tolérantes aux erreurs sont plus flexibles puisqu'il devient possible de "violer" les contraintes précédentes sur les attributs ou la structure. Dans cette dernière famille, la distance d'édition du graphe (GED) est une méthode couramment utilisée qui cherche à minimiser la distance ou la mesure de dissimilarité induite par une série d'opérations d'édition appliquées à un graphe G_1 pour le transformer en un graphe G_2 . Habituellement, les opérations d'édition autorisées sont la substitution, la suppression, l'insertion de sommets ou d'arêtes, avec un coût spécifique associé à chaque opération pouvant dépendre des attributs sur les noeuds et arcs. A nouveau, deux catégories principales de méthodes de résolution existent, les méthodes exactes cherchant la solution optimales (l'appariement de coût minimum) et les méthodes heuristiques cherchant un bon appariement sans garanti qu'il soit optimal. Malheureusement, les méthodes exactes comme A^* [1] ont un temps de calcul élevé. Par contre, les méthodes heuristiques comme A^* -Beam Search [8] ont un temps de calcul plus faible avec une qualité de solution acceptable. De plus, les méthodes exactes semblent être inexploitable pour résoudre le problème d'AG avec des graphes de grande taille. Par conséquent, de plus en plus d'approches heuristiques sont développées.

Selon [12], le problème du AG multivoques est le plus général car il permet à un sommet dans un graphe de correspondre à plus d'un sommet dans l'autre graphe. Dans certaines application, notamment en segmentation d'images, une partie dans un objet (une région) peut correspondre à plusieurs parties dans l'autre objet (plusieurs régions sur-segmentées) [5], [7]. En outre, dans les problèmes de reconnaissance de formes, des distorsions du graphe peuvent se produire, et des techniques de mise en correspondance de graphes tolérantes aux erreurs doivent être utilisées pour permettre l'association de noeuds (arêtes) même s'ils ne sont pas absolument similaires [8]. Toutes ces situations de correspondance sont des cas particuliers d'appariements multivoques. C'est pourquoi nous avons cherché à contribuer à apporter une solution pour le résoudre.

Selon la littérature, nous avons trouvé quelques travaux intéressants traitant du problème de l'AG multivoque. Dans [5], les auteurs appliquent la technique d'AG basée sur la GED pour le problème d'identification des diatomées. Cependant, seuls la fusion et la division de noeuds sont possibles en plus des opérations classiques sans tenir compte des implications sur les arcs. Dans [6], les auteurs proposent également une approche basée sur la GED avec fusion de noeuds pour évaluer la similarité entre des images. Comme dans les travaux précédents, les implications de la fusion de noeuds sur les mises en correspondance entre

arcs ne sont toujours pas claires. En revanche, dans les travaux [6], [10], les auteurs spécifient la fusion des arcs lorsqu'il y a fusion des noeuds. Cependant, les résultats expérimentaux ne sont pas comparés aux autres méthodes. Dans [7], une approche d'AG non-bijectif est utilisée pour rechercher la correspondance entre l'image originale et ses sur-segmentations. A cause du contexte, les auteurs supposent qu'il n'y a que des substitutions de noeuds et d'arcs. De plus, il n'est pas clair où sont la fusion et la division de noeuds et d'arcs. Dans [9], les auteurs proposent une mesure de similarité générale pour l'AG multivoques en employant l'optimisation par colonies de fourmis (OCF) couplée à une recherche tabou réactive pour résoudre le problème. Cependant, la mesure proposée est destinée uniquement aux graphes avec des attributs symboliques ce qui restreint le champ d'application (méthode non tolérante aux distorsions sur les attributs). Dans [14], les auteurs abordent le problème d'AG multivoques comme une extension de la GED appelée GED étendue (e-GED). Une formulation de la e-GED basée sur le concept de matrice de coûts est définie. Une procédure de définition des coûts pour toutes les opérations d'édition possibles sur les noeuds et les arcs (tolérantes aux distorsion sur les attributs) est décrite. Cependant, les temps de calcul impliqués par cette méthode deviennent très rapidement prohibitifs.

En poursuivant le travail [14], nous proposons une 2e formulation du problème. Les coûts associés aux opération de fusion sur les noeuds et les coûts des opérations d'édition sur les arcs induits par les opérations d'édition effectuées sur les noeuds sont intégrés globalement dans une matrice de coûts incorporant à la fois les caractéristiques visuels (attributs/étiquettes sur les noeuds et arcs) et topologiques (structures des graphes). Nous appliquons également les méthodes basées sur les OCF, c'est-à-dire Max-Min Ant System (MMAS), pour résoudre le problème. Nous améliorons les performances de MMAS en réduisant son temps de calcul à l'aide d'une stratégie de recherche locale dans un voisinage défini. Le nombre de candidats à explorer dans l'espace de recherche correspondant à la mise en correspondance de 2 graphes est ainsi considérablement réduits sans trop dégrader la qualité de la solution trouvée. L'efficacité de l'approche proposée est présentée à travers de différents expérimentations réalisé pour le problème de reconnaissance, comparaison d'images de symboles. Les résultats montrent une amélioration significative du temps d'exécution de MMAS tout en réservant une bonne qualité des solutions produites.

La structure de l'article est organisée comme suit. Tout d'abord, la nouvelle formulation du problème de e-GED est présentée dans la section 2. Deuxièmement, la méthode de construction de la matrice des coûts pour e-GED est détaillée dans la section 3. Troisièmement, l'algorithme MMAS intégrant une recherche locale sont spécifiés dans la section 4. Ensuite, l'efficacité de l'approche proposée est démontrée dans la section 5. Enfin, la conclusion résume les principales contributions et propose quelques sugges-

tions pour la suite du travail.

2 Formulation du problème e-GED

2.1 Distance d'édition de graphe

Graphe étiqueté. Un graphe étiqueté est défini par un tuple $G = (V, E, \mu, \xi)$ où :

- V, E sont des ensembles de sommets (nœuds) et d'arêtes (arcs), respectivement,
- l_V, l_E sont des ensembles d'étiquettes de sommets et d'arêtes, respectivement,
- $\mu : V \mapsto l_V$: fonction qui assigne des étiquettes aux sommets
- $\xi : E \mapsto l_E$: fonction qui assigne des étiquettes aux arêtes.

Chemin d'édition. Un chemin d'édition est une séquence d'opérations d'édition (ed_i) afin de transformer un graphe initial en un graphe final, noté $\lambda(G_1, G_2) = \{ed_i\}$. Un chemin d'édition valide doit respecter les conditions suivantes : 1) la suppression d'un sommet implique la suppression des arêtes qui lui sont liées ; 2) l'insertion d'une arête n'est autorisée que si les deux sommets existent déjà ; 3) l'insertion d'une arête ne doit pas créer plus d'une arête entre deux sommets (boucles) [8].

Distance d'édition de graphes. Une distance d'édition de graphes (GED) est une mesure de dissimilarité entre deux graphes $G_1 = (V_1, E_1)$, et $G_2 = (V_2, E_2)$ et est définie par :

$$d_{min}(G_1, G_2) = \min_{\lambda \in \Theta(G_1, G_2)} \sum_{ed_i \in \lambda} c(ed_i), \quad (1)$$

où $\Theta(G_1, G_2)$ est l'ensemble de tous les chemins d'édition valides λ entre G_1 et G_2 , $c(ed_i)$ est le coût de chaque opération d'édition ed_i [8].

Les opérations d'édition classiques sont indiquées dans le Tableau 1. Le coût de chaque opération est défini en fonction des étiquettes des nœuds ou des arêtes.

2.2 La formulation du problème e-GED

Distance d'édition de graphe étendue (e-GED) est également une mesure de dissimilarité dérivée des coûts des opérations d'édition. Pour considérer l'appariement multivoque, nous ajoutons des opérations de division et de fusion.

Division et de fusion. Étant donné deux graphes étiquetés $G_1 = (V_1, E_1, \mu_1, \xi_1)$ and $G_2 = (V_2, E_2, \mu_2, \xi_2)$, nous définissons :

- fusionner l'ensemble $S_{mer} = \{u_i \in V_1, i \geq 2\}$ à $v \in V_2$ est dénoté $S_{mer} \rightarrow v$
- diviser $u \in V_1$ dans l'ensemble $S_{spl} = \{v_j \in V_2, j \geq 2\}$ est dénoté $u \rightarrow S_{spl}$

Ces deux opérateurs sont également mentionnés dans le Tableau 1. En général, la fusion et la division des sommets peuvent entraîner la division et la fusion des arêtes, ce qui sera discuté plus précisément lorsque le coût de chaque opération sera détaillé.

Opération	Notation	Notation de la fonction de coût	GED	e-GED
Substitution de nœuds	$u \rightarrow v$	$c(u \rightarrow v)$	✓	✓
Suppression d'un nœud	$u \rightarrow \varepsilon$	$c(u \rightarrow \varepsilon)$	✓	✓
Insertion d'un nœud	$\varepsilon \rightarrow v$	$c(\varepsilon \rightarrow v)$	✓	✓
Substitution des arcs	$e_1 \rightarrow e_2$	$c(e_1 \rightarrow e_2)$	✓	✓
Suppression d'un arc	$e_1 \rightarrow \varepsilon$	$c(e_1 \rightarrow \varepsilon)$	✓	✓
Insertion d'un arc	$\varepsilon \rightarrow e_2$	$c(\varepsilon \rightarrow e_2)$	✓	✓
Fusion de nœuds	$S_{mer} \rightarrow v$	$c(S_{mer} \rightarrow v)$		✓
Division d'un nœud	$u \rightarrow S_{spl}$	$c(u \rightarrow S_{spl})$		✓

TABLE 1 – Disponibilité des opérations d'édition pour GED et e-GED (avec $u \in V_1, v \in V_2, e_1 \in E_1, e_2 \in E_2, \varepsilon$ le sommet ou l'arête virtuelle. S_{mer} et S_{spl} sont des sous-ensembles de V_1 et V_2 définis dans la précédente définition.

3 Matrices de coûts pour e-GED

3.1 Définition de la matrice des coûts pour les opérations sur les nœuds

Inspiré par [13], une matrice de coûts pour e-GED est également construite avec cinq types d'opérations d'édition. Ces opérations sont placées dans des blocs séparés avec leurs coûts correspondants. Formellement, étant donné deux graphes attribués G_1, G_2 comme ci-dessus, nous avons $n = |V_1|$ et $m = |V_2|$. Soit $P_{mer}^k = \{S_{mer}^i\}$ l'ensemble de toutes les possibilités de fusion de nœuds dans G_1 donc $h = |P_{mer}^k|$, et $P_{spl}^k = \{S_{spl}^j\}$ l'ensemble de toutes les possibilités de division de nœuds dans G_2 donc $l = |P_{spl}^k|$. k est un paramètre qui décrit la taille maximale des ensembles S_{mer}^i et S_{spl}^j , $k \geq 2$. La matrice des coûts est présentée comme suit.

$$\mathbf{C} = \begin{matrix} & \begin{matrix} 1 & \dots & m & \varepsilon & 1 & \dots & l \end{matrix} \\ \begin{matrix} 1 \\ \vdots \\ n \\ \varepsilon \\ 1 \\ \vdots \\ h \end{matrix} & \left(\begin{array}{cccccc} c_{1,1} & \dots & c_{1,m} & c_{1,\varepsilon} & c_{1,S_{spl}^1} & \dots & c_{1,S_{spl}^l} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & \dots & c_{n,m} & c_{n,\varepsilon} & c_{n,S_{spl}^1} & \dots & c_{n,S_{spl}^l} \\ \hline c_{\varepsilon,1} & \dots & c_{\varepsilon,m} & 0 & \infty & \dots & \infty \\ c_{S_{mer}^1,1} & \dots & c_{S_{mer}^1,m} & \infty & \infty & \dots & \infty \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{S_{mer}^h,1} & \dots & c_{S_{mer}^h,m} & \infty & \infty & \dots & \infty \end{array} \right) \end{matrix} \quad (2)$$

où $c_{i,j}$ représente le coût d'une substitution de nœud (avec $(i,j) \in \{1 \dots n\} \times \{1 \dots m\}$), $c_{i,\varepsilon}$ représente le coût d'une suppression d'un nœud, $c_{\varepsilon,j}$ représente le coût d'une insertion d'un nœud, $c_{i,S_{spl}}$ représente le coût d'une division d'un nœud et $c_{S_{mer},j}$ représente le coût d'une fusion de nœuds.

D'après l'équation (2), la matrice des coûts $\mathbf{C}$ n'est pas une matrice carrée comme dans le cas GED en raison des réductions de dimension sur le bloc de suppression $((n \times n) \rightarrow (n \times 1))$ et d'insertion $((m \times m) \rightarrow (1 \times m))$. Ceci a pour but d'accélérer le temps de calcul d'un solveur MMAS tout en préservant la propriété d'une matrice de coût GED. De plus, en introduisant des opérations de division et de fusion, la taille de la matrice $\mathbf{C}$ augmente avec h lignes et l colonnes. h et l sont fortement influencés par k . Plus k est grand, plus les valeurs de h, l sont élevées. Par conséquent, la taille de $\mathbf{C}$ va augmenter de manière significative. Ainsi, le choix du nombre de k sera très important, surtout pour de grands graphes. Habituellement, le para-

mètre k sera dépendant du problème traité et basé sur la connaissance d'experts.

3.2 Définition des coûts pour les opérations sur les arcs

Dans la matrice (2), seuls les coûts des opérations de nœuds sont présentés. Pour enrichir $\mathbf{C}$, nous intégrons le coût estimé des opérations sur les arêtes liés à chaque opération sur un nœud. Le coût engendré par les opérations sur les arcs doit être estimé pour chaque opération sur un nœud. Le détail est donné ci-dessous.

- Pour la substitution de nœuds $u_i \rightarrow v_j$, deux ensembles d'arêtes incidentes de u_i et v_j sont calculés, appelés respectivement E_{u_i} et E_{v_j} . Ensuite, une matrice carrée des coûts des arcs $\mathbf{C}_e$ est construite à partir de E_{u_i} et E_{v_j} en se basant sur les fonctions de coût des opérations sur les arcs dans le Tableau 1. La taille de $\mathbf{C}_e$ est $(|E_{u_i}| + |E_{v_j}|) \times (|E_{u_i}| + |E_{v_j}|)$. Enfin, l'algorithme de Munkres est appliqué à $\mathbf{C}_e$ pour trouver la somme minimale des coûts d'édition des arcs. [13], ou $c_{i,j} \leftarrow c(u_i \rightarrow v_j) + \text{Munkres}(\mathbf{C}_e)$.
- Pour la suppression d'un nœud, la suppression d'un nœud u_i entraînera la suppression de toutes ses arêtes adjacentes, ou $c_{i,\varepsilon} \leftarrow c(u_i \rightarrow \varepsilon) + \sum_{e \in E_{u_i}} c(e \rightarrow \varepsilon)$.
- Pour l'insertion d'un nœud, l'insertion d'un nœud v_j entraînera l'insertion de tous les arcs adjacents, ou $c_{\varepsilon,j} \leftarrow c(\varepsilon \rightarrow v_j) + \sum_{e \in E_{v_j}} c(\varepsilon \rightarrow e)$.
- Pour la fusion de nœuds $S_{mer} \rightarrow v$, on calcule d'abord deux ensembles d'arêtes incidentes aux nœuds de S_{mer} et v , désignés respectivement par $E_{S_{mer}}$ et E_v . Considérons E_{loop} comme l'ensemble des arêtes reliant les nœuds $u_i \in S_{mer}$, nous avons $E'_{S_{mer}} = E_{S_{mer}} \setminus E_{loop}$. Ensuite, une matrice de coût d'arête $\mathbf{C}_e$ pour deux ensembles $E'_{S_{mer}}$ et E_v est construite de la même manière que pour la substitution de nœuds. L'algorithme de Munkres est également utilisé pour trouver le coût minimum pour $E'_{S_{mer}}$ et E_v . Enfin, le coût total de la fusion des nœuds est le suivant $c_{S_{mer},v} \leftarrow c(S_{mer} \rightarrow v) + \text{Munkres}(\mathbf{C}_e) + \sum_{e \in E_{loop}} c(e \rightarrow \varepsilon)$.
- Pour la division d'un nœud $u_i \rightarrow S_{spl}$, les étapes de calcul du coût des arcs sont similaires à la fusion des nœuds : $c_{u,S_{spl}} \leftarrow c(u \rightarrow S_{spl}) + \text{Munkres}(\mathbf{C}_e) + \sum_{e \in E_{loop}} c(\varepsilon \rightarrow e)$.

3.3 Exemple illustratif

Nous donnons ici un exemple du calcul du coût des opérations sur les nœuds et les arêtes. Supposons que G_1 et G_2 soient deux graphes non étiquetés comme dans la Fig. 1, et que chaque nœud puisse être fusionné ou divisé au maximum en deux autres nœuds ($k = 2$). Supposons que les nœuds ayant une arête entre eux puissent être considérés comme des candidats à la fusion ou à

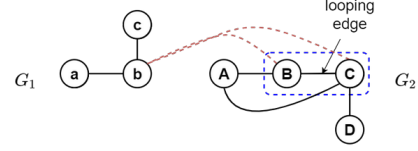


FIGURE 1 – Un graphe illustratif avec un chemin partiel $\lambda = \{b \rightarrow B, C\}$.

la division. En d'autres termes, $P_{mer}^k = \{ab, bc\}$, et $P_{spl}^k = \{AB, AC, BC, CD\}$ sont des ensembles de fusion de nœuds et de division de nœuds dans G_1 et G_2 , respectivement. Nous avons $n = |V_1| = 3$ et $m = |V_2| = 4$. Toutes les fonctions de coût sont définies comme dans le Tableau 2.

Operation	Coût d'édition nœud	Coût d'édition arc
Substitution	$c(u \rightarrow v) = 1$	$c(e_1 \rightarrow e_2) = 1$
Insertion	$c(\varepsilon \rightarrow v) = 2$	$c(\varepsilon \rightarrow e_2) = 2$
Suppression	$c(u \rightarrow \varepsilon) = 10$	$c(e_1 \rightarrow \varepsilon) = 10$
Fusion	$c(S_{mer} \rightarrow v) = 10$	$c(e \rightarrow \varepsilon) = 10, e \in E_{loop}$
Division	$c(u \rightarrow S_{spl}) = 1$	$c(\varepsilon \rightarrow e) = 0, e \in E_{loop}$

TABLE 2 – Fonctions de coût pour les opérations d'édition dans l'exemple.

Avec l'appariement partiel $\lambda = \{b \rightarrow B, C\}$ dans la Fig. 1, nous devons calculer les ensembles d'arêtes incidentes vers les nœuds impliqués comme suit.

- arcs incidents à b : $E_b = \{(a, b), (b, c)\}$
- arcs en boucle : $E_{loop} = \{(B, C)\}$
- arcs incidents à B, C :
 $E_{BC} = \{(A, B), (A, C), (B, C), (C, D)\}$
- arcs incidents à B, C , sauf E_{loop} :
 $E'_{BC} = \{(A, B), (A, C), (C, D)\}$

Ensuite, la matrice des coûts des arcs $\mathbf{C}_e$ pour E_b et E_{BC} est calculée selon la matrice (3). Le coût total pour le chemin partiel $\lambda = \{b \rightarrow B, C\}$ est :

$$c_{b,BC} = c(b \rightarrow B, C) + \text{Munkres}(\mathbf{C}_e) + c(\varepsilon \rightarrow BC) = 1 + 4 + 0 = 5.$$

$$\mathbf{C}_e = \begin{matrix} & \begin{matrix} ab & bc \end{matrix} \\ \begin{matrix} \varepsilon_{AB} \\ \varepsilon_{AC} \\ \varepsilon_{CD} \end{matrix} & \begin{pmatrix} c_{ab,AB} & c_{ab,AC} & c_{ab,CD} & c_{ab,\varepsilon} & c_{bc,\varepsilon} \\ c_{bc,AB} & c_{bc,AC} & c_{bc,CD} & \infty & c_{bc,\varepsilon} \\ c_{\varepsilon,AB} & \infty & \infty & 0 & 0 \\ \infty & c_{\varepsilon,AC} & \infty & 0 & 0 \\ \infty & \infty & c_{\varepsilon,CD} & 0 & 0 \end{pmatrix} \end{matrix} \quad (3)$$

$$= \begin{pmatrix} 1 & 1 & 1 & 10 & \infty \\ 2 & \infty & \infty & 0 & 0 \\ \infty & 2 & \infty & 0 & 0 \\ \infty & \infty & 2 & 0 & 0 \end{pmatrix}$$

4 Résolution de e-GED avec une variante de MMAS

4.1 Schéma algorithmique

Dans ce travail, nous utilisons Max-Min Ant System (MMAS) qui est une variante de l'optimisation par colonies de fourmis (OCF) [2]. Une colonie de fourmis est utilisée pour générer des solutions au problème considéré de manière parallèle. À chaque itération, chaque fourmi construit un appariement complet basé sur les probabilités de transition. Un graphe de construction, noté G_{ants} ,

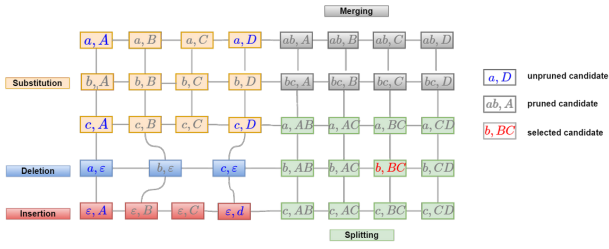


FIGURE 2 – Graphe de construction représentant tous les appariements possibles de nœuds des deux graphes de la Fig. 1. Certaines arêtes sont présentées pour faciliter la lecture mais tous les nœuds sont entièrement connectés.

est utilisé pour connaître les possibilités de mouvement d'une fourmi. En d'autres termes, G_{ants} est l'espace de recherche qui montre toutes les possibilités d'appariement de nœuds entre deux graphes. Dans ce travail, G_{ants} est considéré comme un graphe complet non orienté où chaque nœud de G_{ants} représente une opération d'édition entre les nœuds de G_1 et G_2 . Les arcs de G_{ants} visent à construire de manière incrémentale le meilleur chemin d'édition. Les phéromones des fourmis naturelles correspondent ici à des probabilités qui sont partagées entre les fourmis pour construire une solution. Une valeur initiale de phéromone est déposée sur chaque sommet de G_{ants} et la valeur de la phéromone change en fonction de la performance du meilleur appariement trouvé et des mécanismes de mise à jour.

4.2 Graphe de construction

Un graphe de construction G_{ants} correspond à un espace de recherche dans lequel on trouve tous les appariements de nœuds possibles (y compris les cas étendus) entre deux graphes. En d'autres termes, G_{ants} est construit pour représenter tous les chemins d'édition à explorer lors de la résolution du problème e-GED défini dans la section précédente. La Fig. 2 montre le graphe de construction construit à partir des graphes de la Fig. 1. Les candidats pour chaque opération d'édition sont les produits cartésiens des ensembles de nœuds comme dans le Tableau 3. Remarquons que la possibilité pour une fourmi de se déplacer sur G_{ants} dépend du chemin d'édition partiel actuel. Par exemple, si le chemin d'édition partiel est $\lambda = \{b \rightarrow B, C\}$, tous les candidats contenant un de ces nœuds sont élagués (nœuds gris). Cela signifie que la fourmi actuelle ne peut pas se déplacer sur ces positions jusqu'à ce que le chemin d'édition soit complet.

Opération	Substitution	Suppression	Insertion	Fusion	Division
Candidates	$V_1 \times V_2$	$V_1 \times \varepsilon$	$\varepsilon \times V_2$	$P_{mer}^k \times V_2$	$V_1 \times P_{spl}^k$

TABLEAU 3 – Candidats pour chaque opération d'édition dans le graphe de construction, où V_1, V_2 sont des ensembles de nœuds dans G_1, G_2 , respectivement, ε est un nœud virtuel, P_{mer}^k, P_{spl}^k sont des ensembles de nœuds de fusion et de division, respectivement.

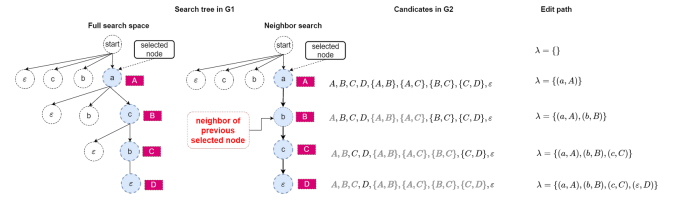


FIGURE 3 – Arbre de recherche dans G_1 sans et avec la stratégie de recherche par voisinage, les nœuds gris indiquent les nœuds élagués dans G_2 pour le moment.

4.3 Stratégie de recherche par voisinage

Construction d'un appariement complet À chaque itération, chaque fourmi construit un chemin d'édition complet. Elle commence par un chemin d'édition vide $\lambda = \{\}$, puis elle ajoute un nouveau candidat $v_i \in G_{ants}$ à λ en fonction de la probabilité de transition, jusqu'à ce que λ contienne tous les nœuds des deux graphes. La probabilité de transition est calculée sur la base de deux facteurs : l'heuristique et la phéromone. Dans cet article, l'heuristique est dérivée de la matrice des coûts C et la phéromone est posée sur le graphe de construction G_{ants} . Si τ_{v_i}, η_{v_i} sont respectivement les valeurs de la phéromone et de l'heuristique du candidat v_i , alors la probabilité de transition de v_i est :

$$Pr_{v_i} = \frac{[\tau_{v_i}]^\alpha \times [\eta_{v_i}]^\beta}{\sum_{j=1}^{|cand|} [\tau_{v_j}]^\alpha \times [\eta_{v_j}]^\beta} \quad (4)$$

Stratégie de recherche par voisinage De plus, nous considérons une stratégie de recherche par voisinage pour réduire le temps de calcul de MMAS. Cela signifie qu'à partir d'un nœud précédemment sélectionné dans G_1 , nous nous concentrons uniquement sur ses voisins pour trouver les prochains candidats. Si aucun voisin n'est trouvé, un choix aléatoire est effectué. Cette recherche par stratégie locale peut passer à côté de bonnes solutions, mais elle prend moins de temps. Il faut aussi noter que dans le cas d'un graphe complet, cette stratégie est inutile. Une démonstration de l'amélioration est donnée dans la Fig. 3.

4.4 Mise-à-jour des phéromones

Le processus de mise à jour des phéromones est appliqué à chaque nœud de G_{ants} après que toutes les fourmis aient construit leur solution à cette itération, y compris le renforcement et l'évaporation. Le renforcement est effectué uniquement sur la meilleure solution de l'itération en cours (λ_{itbest}). L'évaporation est effectuée pour tous les nœuds selon le taux d'évaporation $\rho \in [0, 1]$. Ensuite, l'ajustement des phéromones suivra pour contraindre toutes les valeurs des phéromones dans l'intervalle $[\tau_{min}, \tau_{max}]$ [2].

$$\tau_{v_i} = (1 - \rho) * \tau_{v_i} + \Delta_{v_i}, \quad (5)$$

$$\Delta_{v_i} = \begin{cases} \frac{1}{1+c(\lambda_{itbest})} & \text{si } v_i \in \lambda_{itbest} \\ 0 & \text{sinon} \end{cases}$$

5 Expérimentations

Toutes les expériences sont implémentées en Python 3.8.5 et exécutées sur Windows 10 Intel(R) Core(TM) i7-8750 CPU @ 2.20 GHz, RAM de 16.0 Go. À partir de maintenant, nous appelons l'approche proposée MMAS-e-GED-voi.

L'algorithme MMAS est utilisé pour rechercher la meilleure correspondance entre deux graphes. L'espace de recherche est énorme et le problème combinatoire sous-jacent ne peut pas être résolu exactement en un temps raisonnable [10]. Cependant, il est nécessaire d'ajuster les paramètres de MMAS pour qu'ils soient bien adaptés au problème.

5.1 Ensemble de données utilisé

Pour l'expérience, nous utilisons une base de données publique construite à partir d'images de symboles SE-SYD. Cette base contient des symboles architecturaux et électriques. Les symboles déformés sont créés en ajoutant du bruit dans les images pour casser une ligne en plusieurs lignes ou en faisant pivoter ou en modifiant l'échelle des symboles. Ensuite, un algorithme de vectorisation d'images binaires a été utilisé pour transformer les images en graphes. Dans ces graphes, les nœuds représentent les lignes/tracés manuscrits et les arêtes représentent les relations topologiques estimées entre les lignes. Les attributs/étiquettes des nœuds sont la longueur relative (l) des lignes qui est normalisée en fonction de la ligne la plus longue dans chaque graphe. Les attributs des arêtes sont l'angle relatif ($\theta \in [0, 180]$) et le type de relation (rel). Le type de relation rel est un attribut symbolique qui peut prendre les valeurs suivantes : jonction en T (T), parallèle (P), successif (S), jonction en L (L) ou intersection (X) [11]. Nous donnons un exemple dans la Fig. 4¹. La rela-

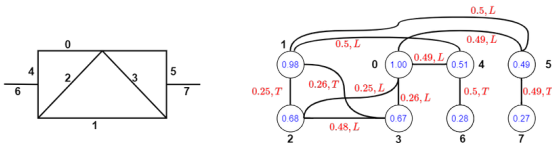


FIGURE 4 – Le symbole (à gauche) et sa représentation sous forme de graphe (à droite) avec les attributs des nœuds (bleu) et des arêtes (rouge).

tion S apparaît lorsque les tracés manuscrits sont bruités. Ainsi, nous définissons que les nœuds qui sont connectés par des arêtes avec des relations S seront des candidats pour la fusion ou la division. De plus, nous fixons le nombre maximum de fusion/division à $k = 3$ pour cet ensemble de données.

5.2 Définition des fonctions de coût pour les opérations d'édition

Généralement, la définition des fonctions de coût se base sur des connaissances expertes (application visée et dé-

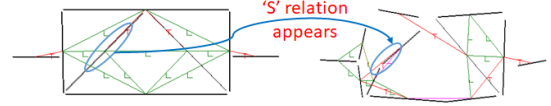


FIGURE 5 – La relation S apparaît lorsque les tracés (à gauche) sont divisés en plusieurs segments dans le symbole déformé (à droite).

formations induites par l'algorithme de production des graphes à partir des données initiales). Ici, les données initiales sont des images binaires qui vont être soumises à un algorithme de vectorisation puis d'analyse topologique pour produire des graphes représentant le contenu des images de symboles.

Pour les opérations sur les nœuds, le coût est proportionnel à l'attribut de longueur (l). Cela signifie que la suppression/insertion d'une longue ligne devrait coûter plus cher que l'introduction d'une ligne plus courte. La substitution de deux lignes de longueur équivalente devrait être zéro. De même, la fusion de lignes coûtera moins chère si cela génère une ligne de longueur adéquate. Les opérations sur les arêtes, le coût dépend du type de relation (rel) et de l'angle relatif (θ). Les valeurs renvoyées par les fonctions de coût, aussi bien pour les nœuds que pour les arêtes sont définies de sorte à être toujours comprises entre $[0, 1]$ pour être compatibles entre elles (normalisation des valeurs renvoyées par les fonctions de coûts). Les détails des fonctions de coût sont dans le Tableau 4.

Node operation cost	Edge operation cost
$c(u \rightarrow v) = l_u - l_v $	$c(e_1 \rightarrow e_2) = \begin{cases} \theta_{e_1} - \theta_{e_2} & \text{if } rel_{e_1} = rel_{e_2} \\ 1 & \text{otherwise} \end{cases}$
$c(\varepsilon \rightarrow v) = l_v$	$c(\varepsilon \rightarrow e_2) = 1$
$c(u \rightarrow \varepsilon) = l_u$	$c(e_1 \rightarrow \varepsilon) = 1$
$c(S_{mer} \rightarrow v) = \sum_{u_i \in S_{mer}} l_{u_i} - l_v$	$c(e \rightarrow \varepsilon) = 0, e \in E_{loop}$
$c(u \rightarrow S_{spl}) = l_u - \sum_{v_i \in S_{spl}} l_{v_i}$	$c(\varepsilon \rightarrow e) = 0, e \in E_{loop}$

TABLE 4 – Fonctions de coût des opérations d'édition pour l'ensemble de données.

5.3 Réglage des paramètres du MMAS

Dans cette partie, nous étudions les impacts des principaux paramètres de MMAS et de la recherche locale sur les résultats (appariements produits). Chaque étude de paramètres est exécutée sur 30 itérations et les résultats moyens sont présentés. Pour cette expérimentation, nous avons fixé le nombre de fourmis à 5 ($nb_{ants} = 5$) et $[\tau_{min}, \tau_{max}] = [0.1, 2.0]$. La Fig. 6 montre l'influence des paramètres α, β, ρ sur les coûts finaux. Sur la base de ces résultats, nous avons choisi $\alpha = 1, \beta = 3, \rho = 0.01$ et une recherche locale de type 3-opt pour les expériences ultérieures.

5.4 Analyse de la qualité de l'appariement

Dans cette section, nous évaluons la qualité de l'appariement en ce qui concerne la correspondance entre les sous-parties ou la correspondance entre les nœuds. Le résultat

¹ lien : <http://www.rfai.lifat.univ-tours.fr/PublicData/ExGED/home.html>

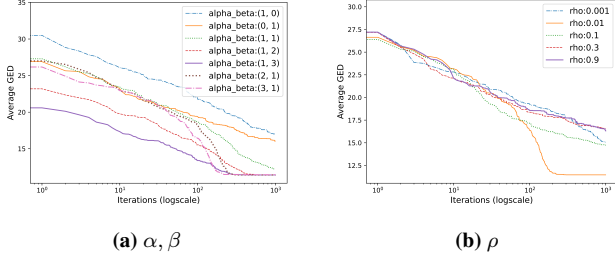


FIGURE 6 – Influences des principaux paramètres sur la convergence de MMAS vers e-GED (avec 5 fourmis, 300 itérations, $[\tau_{min}, \tau_{max}] = [0.1, 2.0]$).

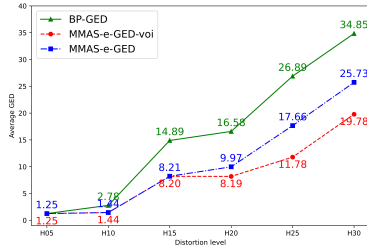


FIGURE 7 – Coûts moyens de BP-GED, MMAS-e-GED-voi et MMAS-e-GED en fonction des niveaux de distorsion.

de l'approche est comparé à l'une des méthodes bipartites pour GED [13] (appelée BP-GED) et à celle de [14] (dénotee MMAS-e-GED). Nous considérons qu'un appariement est raisonnable s'il fait correspondre correctement les différents tracés constituant deux symboles (symboles d'origine VS symbole bruité). La correspondance est vérifiée visuellement / manuellement. Nous prenons 5 modèles et 6 niveaux de distorsions pour chaque modèle pour cette expérience.

La Fig. 7 montre le coût moyen donné par trois méthodes en fonction des niveaux de distorsion. Nous remarquons que le coût est proportionnel au niveau de distorsion en général, un coût plus élevé pour une distorsion plus importante. Dans tous les cas, la BP-GED obtient une dissimilarité plus élevée que les deux méthodes basées sur e-GED. Cela confirme l'adéquation de e-GED à ce type de bruit par rapport à la GED.

Les différences de coût entre les méthodes reflètent les appariements obtenus. Plus précisément, les méthodes basées sur e-GED peuvent trouver des correspondances plus appropriées que BP-GED grâce à la fusion et au fractionnement. Une démonstration est donnée dans la Fig. 8 et le Tableau 5. De plus, les résultats donnés par le MMAS-e-GED-voi sont assez compétitifs avec le MMAS-e-GED, ce qui confirme que la méthode proposée prend moins de temps tout en conservant de bons résultats par rapport à MMAS-e-GED. Par exemple, le MMAS-e-GED-voi prend seulement 149.425s pour obtenir les résultats de la Fig. 7 plutôt que 705.048s de MMAS-e-GED pendant 500 itérations.

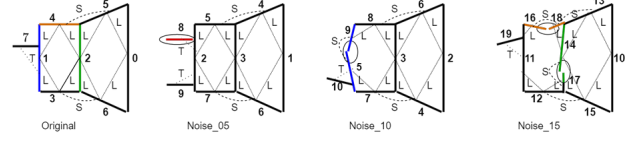


FIGURE 8 – Représentation des nœuds (en gras) et des arcs (en pointillés) sur le symbole 032 et ses niveaux de distorsion. L'ellipse indique la position du bruit, la même position de bruit étant illustrée avec la même couleur.

Méthodes	Niveau 5	Niveau 10	Niveau 15
BP-GED	$\varepsilon \rightarrow 9$	$1 \rightarrow 5; \varepsilon \rightarrow 9$	$2 \rightarrow 14; 3 \rightarrow 17; 0 \rightarrow 16; 6 \rightarrow 18$
Basé sur e-GED	$\varepsilon \rightarrow 8$	$1 \rightarrow \{5, 9\}$	$2 \rightarrow \{14, 17\}; 4 \rightarrow \{16, 18\}$

TABLE 5 – Correspondance entre les nœuds aux positions déformées entre le modèle 032 et ses bruits dans la Fig. 8 donnée par les méthodes basées sur BP-GED et e-GED.

5.5 Problème de classification

Dans la section précédente, nous avons souligné l'intérêt d'introduire la fusion et la division de sommets et d'arcs pour bien reconnaître les symboles bruités en termes d'analyse de la qualité de la mise en correspondance (entre sous-parties d'objets ou images) sur de petits jeux de données. Dans cette partie, nous essayons d'indiquer l'efficacité de la méthode proposée pour un jeu de données plus important dans un problème de classification d'images (représentés par des graphes). Pour ce faire, nous concevons l'expérience comme suit. Nous sélectionnons au hasard 20 modèles dans l'ensemble de données de symboles comme ensemble d'entraînement (Fig. 9) et environ 20 instances bruitées de chaque modèle comme ensemble de test. Par conséquent, la taille de l'ensemble d'entraînement ($\#train = 20$) est beaucoup plus petite que celle de l'ensemble de test ($\#test = 418$). Nous pensons que la méthode proposée peut obtenir un bon taux de classification avec un faible ensemble d'entraînement. Les performances de la méthode MMAS-e-GED-voi sont mesurées par rapport à celles de BP-GED [13], algorithme de glouton pour e-GED (Glouton-e-GED), MMAS-e-GED [14], et MMAS-GED. Le paramétrage de cette expérience est décrit dans le Tableau 6.

Paramètre	(α, β)	$maxiter$	nb_{ants}	ρ	(τ_{min}, τ_{max})	k
Valeur	(1,3)	50,100	5	0.01	(0.01,2.0)	2,3

TABLE 6 – Paramétrage du problème de reconnaissance des symboles.

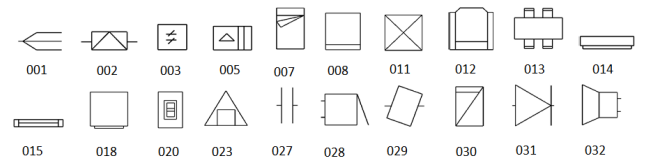


FIGURE 9 – Modèles pour le problème de classification dans l'expérience.

Selon les résultats expérimentaux du Tableau 7, nous observons les points principaux suivants. Premièrement, les approches basées sur MMAS donnent une meilleure précision mais prennent plus de temps que les autres approches (BP-GED, Glouton-e-GED). C'est normal puisque MMAS doit itérer le processus de recherche plusieurs fois pour améliorer la qualité de sa solution. Néanmoins, si l'on examine le temps moyen de construction d'une correspondance, les approches basées sur MMAS sont plus économes en temps. Deuxièmement, entre les approches basées sur GED et celles basées sur e-GED, celles basées sur e-GED sont généralement plus robustes avec un taux de classification plus élevé, en particulier le MMAS-e-GED-voi. Troisièmement, de meilleurs résultats sont obtenus avec $k = 3$ plutôt que $k = 2$ car cela offre des possibilités de mise en correspondance supplémentaires. En revanche, elle exige plus de temps de calcul en raison de l'augmentation de la taille de la matrice des coûts. Enfin, l'MMAS-e-GED-voi prouve son efficacité par rapport au MMAS-e-GED, ce qui confirme notre attente initiale d'équilibre entre la qualité de la solution et le temps de calcul.

Méthode	($k, \text{maxiter}$)	Précision (%)	Temps/appariement(s)	Temps total(s)	#appariement
BP-GED		83.014	0.008	66.621	8360
Glouton-e-GED	(2,1)	74.402	0.120	96.347	8360
	(3,1)	74.880	0.012	100.561	8360
MMAS-e-GED-voi	(2,50)	84.450	0.002	4130.574	2090000
	(3,50)	85.885	0.002	4326.690	2090000
	(2,100)	84.211	0.002	8401.295	4180000
	(3,100)	86.124	0.002	8267.924	4180000
MMAS-e-GED	(2,50)	84.211	0.005	9996.545	2090000
	(3,50)	85.167	0.005	10271.611	2090000
	(2,100)	83.732	0.005	20481.097	4180000
	(3,100)	85.407	0.005	20754.161	4180000
MMAS-GED	(_,50)	83.253	0.004	9451.112	2090000
	(_,100)	83.253	0.004	18683.444	4180000

TABLE 7 – Résultats de classification des symboles donnés par les approches basées sur GED et e-GED, le meilleur résultat de chaque méthode est en gras.

6 Conclusion

Dans cet article, une approche e-GED efficace est proposée pour résoudre un problème d'appariements multivoques de graphes permettant d'appréhender de manière différente la problématique de l'analyse de contenu d'images ou de reconnaissance de formes. Plus précisément, e-GED propose d'ajouter des opérations de fractionnement et de fusion dans le processus de calcul d'une distance d'édition afin d'introduire plus de tolérance aux bruits et distorsions lors de la mise en correspondance des sous-parties d'un objet. La formulation du problème de e-GED est présentée en détail en se basant sur le concept de matrice de coûts. Ensuite, MMAS avec une recherche locale est utilisé pour produire une solutions sous-optimale rapidement. L'efficacité de la proposition est démontrée dans une application de reconnaissance d'images de symboles avec différents niveaux de distorsions. Les résultats démontrent les aspects qualitatifs (qualité et interprétabilité des correspondances générées) et quantitatifs (performance de classification). Ces résultats positifs nous encouragent à appliquer la méthode proposée à d'autres ensembles de données, par exemple, la comparaison de connectomes de cerveau à différentes échelles.

Références

- [1] P. E. HART, N. J. NILSSON et B. RAPHAEL, « A formal basis for the heuristic determination of minimum cost paths, » *IEEE Transactions on Systems Science and Cybernetics*, t. 4, p. 100-107, 1968.
- [2] T. STÜTZLE et H. H. HOOS, « MAX-MIN ant system, » *Future Generation Computer Systems*, t. 16, p. 889-914, 2000.
- [3] A. SHOKOUFANDEH et S. DICKINSON, « A unified framework for indexing and matching hierarchical shape structures, » in *IWVF*, Springer, 2001, p. 67-84.
- [4] A. TEFAS, C. KOTROPOULOS et I. PITAS, « Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication, » *IEEE TPAMI*, t. 23, p. 735-746, 2001.
- [5] R. AMBAUEN, S. FISCHER et H. BUNKE, « Graph edit distance with node splitting and merging, and its application to diatom identification, » in *GbrPR*, Springer, 2003, p. 95-106.
- [6] S. BERRETTI, A. DEL BIMBO et P. PALA, « A graph edit distance based on node merging, » in *ICIVR*, Springer, 2004, p. 464-472.
- [7] M. C. BOERES, C. C. RIBEIRO et I. BLOCH, « A randomized heuristic for scene recognition by graph matching, » in *International Workshop on Experimental and Efficient Algorithms*, Springer, 2004, p. 100-113.
- [8] D. CONTE, P. FOGGIA, C. SANSONE et M. VENTO, « Thirty years of graph matching in pattern recognition, » *IJPRAI*, t. 18, p. 265-298, 2004.
- [9] O. SAMMOUD, C. SOLNON et K. GHÉDIRA, « Ant algorithm for the graph matching problem, » in *EvoCOP*, Springer, 2005, p. 213-223.
- [10] S. BERRETTI, A. D. BIMBO et P. PALA, « Graph edit distance for active graph matching in content based retrieval applications, » *The Open Artificial Intelligence Journal*, t. 1, 2007.
- [11] R. J. QURESHI, J.-Y. RAMEL, H. CARDOT et P. MUKHERJI, « Combination of symbolic and statistical features for symbols recognition, » in *ICSPCN*, IEEE, 2007, p. 477-482.
- [12] S. SORLIN, C. SOLNON et J.-M. JOLION, « A generic graph distance measure based on multivalent matchings, » in *Applied Graph Theory in Computer Vision and Pattern Recognition*, Springer, 2007, p. 151-181.
- [13] K. RIESEN et H. BUNKE, « Approximate graph edit distance computation by means of bipartite graph matching, » *Image and Vision Computing*, t. 27, p. 950-959, 2009.
- [14] K. D. HO, J.-Y. RAMEL et N. MONMARCHÉ, « Multivalent Graph Matching Problem Solved by Max-Min Ant System, » in *S+SSPR*, t. 12644, Springer, 2020, p. 227-237.
- [15] M. STAUFFER, A. FISCHER et K. RIESEN, « Filters for graph-based keyword spotting in historical handwritten documents, » *Pattern Recognition Letters*, t. 134, p. 125-134, 2020.