

A column generation approach for the graph matching problem

A.S. Freire[‡], R. M. Cesar Jr.* and C.E. Ferreira[†]

Instituto de Matemática e Estatística – Universidade de São Paulo – Brazil

*‡ afreire@ime.usp.br; * cesar@ime.usp.br; † cef@ime.usp.br*

Abstract—Graph matching plays a central role in different problems for structural pattern recognition. Examples of applications include matching 3D CAD models, shape matching and medical imaging, to name but a few. In this paper, we present a new integer linear formulation for the problem and employ a combinatorial optimization technique, called “column generation”, in order to solve instances of the problem. We also present computational experiments with generated instances.

I. INTRODUCTION AND MOTIVATION

Graph matching plays a central role in different problems for structural pattern recognition [1], [2]. Examples of applications include matching 3D CAD models [3], shape matching [4], [5] and medical imaging [6], to name but a few. In many important cases, graph matching is applied in order to solve model-based recognition problems such as in image analysis [7]. A model pattern is composed by a set of parts organized by some relations among the parts. A *model graph* G_M represents the model by associating parts to vertices and relations to edges. Input patterns to be recognized are analogously represented by an *input graph* G_I and the pattern recognition task is accomplished by matching G_I to G_M .

In such approaches, the graphs represent appearance and relational features. Appearance features characterize the pattern parts (e.g. gray-level for image segmentation) and are stored in the vertices, while relational features characterize the relations (e.g. distance and orientation between parts) and are stored in the edges. By defining suitable dissimilarity functions measured by comparing input and model features, the graph matching may be modeled as an optimization problem. Variations on these elements (types of graphs and constraints on them, valid features and dissimilarity functions, etc.) lead to different formulations and solutions for the graph matching problem. The literature on these topics is very rich and has been growing intensively in the last 20 years. As far as the matching algorithm is concerned, a large variety of approaches may be found, including combinatorial optimization and relaxation

techniques, expectation maximization (EM), estimation of distribution algorithms (EDAs), genetic algorithms, tree-search and propagation techniques, heuristic-based graph traversing, graph editing and graph labeling based on probabilistic models of attributes. The reader is referred to [7] to a short review with links to the literature. See also [8] for a related approach.

In this paper, we present a new formulation for graph matching as an integer linear programming (ILP) problem. One of the greatest difficulties in such approaches is the big gap between the linear program (LP) optimal solutions and the integer program (IP) optimal solutions. A possible alternative is to develop huge formulations, in order to get tight gaps, and employ advanced techniques to use them in practice. The method presented here is based on a combinatorial optimization technique called *column generation* [9].

This paper is organized as follows. Section II introduces the notation and some definitions required to describe our method. Section III presents the proposed method, while Section IV shows the computational experiments. The paper is concluded with some comments on our ongoing work in Section V.

II. NOTATION AND DEFINITIONS

Let $G_I = (V_I, E_I)$ and $G_M = (V_M, E_M)$ be two undirected connected graphs. We say that a function $\alpha : V_I \rightarrow V_M$ is a (G_I, G_M) -matching. Two cost functions are given, namely $c : V_I \times V_M \rightarrow \mathbb{R}_+$ and $d : E_I \times V_M \times V_M \rightarrow \mathbb{R}_+$, such that c_{ik} is the cost of matching the vertex $i \in V_I$ to the vertex $k \in V_M$, and d_{ij}^{kl} is the cost of matching the edge $ij \in E_I$ to the pair of vertices $(k, l) \in V_M \times V_M$ (for simplicity we use this notation instead of $c(i, k)$ and $d(ij, k, l)$, respectively). The cost of α is calculated as follows: $C(\alpha) = \sum_{i \in V_I} c_{i\alpha(i)} + \sum_{ij \in E_I} d_{ij}^{k\alpha(i)l\alpha(j)}$, where $k = \alpha(i)$ and $l = \alpha(j)$.

Let A_M be the set of all pairs of vertices of V_M for which there is an edge between them, i.e. $A_M = \{(k, l) \in V_M \times V_M \mid kl \in E_M\}$, and let \bar{A}_M be the set of all pairs of different vertices of

V_M for which there is no edge between them, i.e. $\bar{A}_M = \{(k, l) \in V_M \times V_M \mid k \neq l \text{ and } kl \notin E_M\}$. Given a subset $U \subseteq V_I$, we denote by $G_I[U]$ the subgraph of G_I induced by U .

We say that α is *feasible* if it satisfies the following constraints: (*edge cover constraint*) for each edge $ij \in E_I$ we have that $(\alpha(i), \alpha(j)) \notin \bar{A}_M$; and (*connectivity constraint*) for each vertex $k \in V_M$ we have that $G_I[\lambda(k)]$ is connected, where $\lambda(k)$ is the subset of V_I whose elements are matched to k in α (i.e. $\lambda(k) = \{i \in V_I \mid \alpha(i) = k\}$). The graph matching problem (GMP) that we consider here consists of finding a minimum cost feasible (G_I, G_M) -matching.

A *cluster* is a pair (U, k) such that $U \subseteq V_I$, $G_I[U]$ is connected, $k \in V_M$ and $\alpha(i) = k$, for all $i \in U$. We define \mathcal{S} as the set of all possible clusters, i.e. $\mathcal{S} = \{(U, k) \mid U \subseteq V_I \text{ and } k \in V_M\}$. We define \mathcal{S}_{i^*} as the set of all clusters which contain the vertex i (i.e. $\mathcal{S}_{i^*} = \{(U, k) \in \mathcal{S} \mid i \in U\}$) and \mathcal{S}_{*k} as the set of all clusters (U, l) such that $k = l$ (i.e. $\mathcal{S}_{*k} = \{(U, l) \in \mathcal{S} \mid l = k\}$). Given a cluster $S = (U, k)$, the cost of S is calculated as follows: $C_S = \sum_{i \in U} c_{ik} + \sum_{ij \in E_I \mid i, j \in U} d_{ij}^{kk}$.

Given a graph $G = (V, E)$, we denote by $\delta_G(u)$ the vertices adjacent to u (the neighbours of u in G), and $\bar{\delta}_G(u) = V \setminus (\delta_G(u) \cup \{u\})$. When the graph G is implicit in the context we use simply $\delta(u)$ and $\bar{\delta}(u)$.

III. A COLUMN GENERATION APPROACH FOR THE GMP

For each $S = (U, k) \in \mathcal{S}$, let $x_S \in \{0, 1\}$ be such that $x_S = 1$ if and only if $\alpha(i) = k$ for all $i \in U$. For each $ij \in E_I$ and $kl \in E_M$, let $y_{ij}^{kl} \in \mathbb{R}_+$ be such that $y_{ij}^{kl} = 1$ if and only if ij is matched to the edge kl , i.e. $\alpha(i) = k$ and $\alpha(j) = l$ (or $\alpha(i) = l$ and $\alpha(j) = k$). Consider the following formulation, namely (P_{GMP}) , for the GMP.

$$\begin{aligned} \min \quad & \sum_{S \in \mathcal{S}} C_S x_S + \sum_{ij \in E_I} \sum_{kl \in E_M} d_{ij}^{kl} y_{ij}^{kl} \\ \text{s.t.} \quad & \sum_{S \in \mathcal{S}_{i^*}} x_S = 1, \quad \forall i \in V_I. \end{aligned} \quad (1)$$

$$\sum_{S \in \mathcal{S}_{*k}} x_S \leq 1, \quad \forall k \in V_M. \quad (2)$$

$$\sum_{S \in \mathcal{S}_{ik}} x_S + \sum_{S \in \mathcal{S}_{jl}} x_S \leq 1, \quad \forall ij \in E_I, \quad \forall (k, l) \in \bar{A}_M. \quad (3)$$

$$\sum_{S \in \mathcal{S}_{ik}} x_S + \sum_{S \in \mathcal{S}_{jl}} x_S - y_{ij}^{kl} \leq 1, \quad \forall ij \in E_I, \quad \forall (k, l) \in A_M. \quad (4)$$

$$0 \leq y_{ij}^{kl} \leq 1, \quad \forall ij \in E_I \text{ and } kl \in E_M. \quad (5)$$

$$x_S \in \{0, 1\}, \quad \forall S \in \mathcal{S}. \quad (6)$$

Constraints (1) guarantee that each vertex of V_I is matched to exactly one vertex of V_M . Constraints (2) guarantee that $G_I[\lambda(k)]$ is connected for all $k \in V_M$. Constraints (3) guarantee that the edge cover constraint is satisfied. Finally, constraints (4) guarantee that $y_{ij}^{kl} = 1$ if and only if $\alpha(i) = k$ and $\alpha(j) = l$ (or $\alpha(i) = l$ and $\alpha(j) = k$). The objective is to minimize the cost of the matching.

Let (LP_{GMP}) be the linear relaxation of (P_{GMP}) (where the constraints (6) are replaced by $0 \leq x_S \leq 1$, for all $S \in \mathcal{S}$). Since the size of \mathcal{S} increases exponentially in the input size, it is impracticable to solve even the relaxed model (LP_{GMP}) with all variables. Instead, the column generation approach (see Barnhart et al. [9] for an introduction) attempts to include in (LP_{GMP}) only the variables that are “really needed”, and it works as follows: (step 1) let (RLP_{GMP}) be the model (LP_{GMP}) restricted to a small subset of the variables (the restricted model (RLP_{GMP}) must be feasible); (step 2) compute (x^*, y^*) , the solution of (RLP_{GMP}) ; (step 3) check if there is a variable x_S with negative reduced price in (LP_{GMP}) (a variable x_S such that if x_S was included in (RLP_{GMP}) then the objective function would be improved); if there is no such a variable, stop ((x^*, y^*) is an optimal solution to (LP_{GMP})); (step 4) include x_S in (RLP_{GMP}) and return to step 2. Since there are $|E_I| \cdot |E_M|$ variables y , we include in (RLP_{GMP}) all variables y a priori in step 1.

The column generation procedure provides a practicable way to solve (LP_{GMP}) , and it can be used in a *branch-and-price* [9] algorithm in order to solve (P_{GMP}) . The heuristic we propose here consists of two steps: (step 1) solve the model (LP_{GMP}) by column generation; (step 2) let (RP_{GMP}) be the model (P_{GMP}) restricted to the variables included in the column generation procedure; use a MIP (mixed integer programming) solver to solve (RP_{GMP}) . Note that this procedure calculates both an upper bound and a lower bound on the optimal solution, providing a “quality certificate” of the solution found.

Now we concentrate on how to solve the *pricing problem*, i.e. to find a variable with negative reduced price, or to prove that no such variable exists. Observe that the pricing problem can be solved by finding a variable with minimum reduced price. Therefore, we focus on how to find a variable with minimum reduced price.

For each constraint in (1), (2), (3) and (4) of (LP_{GMP}) we associate a dual variable $\psi_i, \phi_k, \varphi_{ij}^{kl}$ and ω_{ij}^{kl} , respectively. Define $\gamma(U, k, l) = \sum_{ij \in E_I: i \in U} \varphi_{ij}^{kl} + \sum_{ij \in E_I: j \in U} \varphi_{ij}^{lk}$ and $\rho(U, k, l) = \sum_{ij \in E_I: i \in U} \omega_{ij}^{kl} + \sum_{ij \in E_I: j \in U} \omega_{ij}^{lk}$, and

consider the dual linear program of (LP_{GMP}):

$$\begin{aligned}
\max \quad & \sum_{i \in V_I} \psi_i - \sum_{k \in V_M} \phi_k - \sum_{\substack{ij \in E_I, \\ (k,l) \in \bar{A}_M}} \varphi_{ij}^{kl} - \sum_{\substack{ij \in E_I, \\ (k,l) \in A_M}} \omega_{ij}^{kl} \\
\text{s.t.} \quad & \sum_{i \in U} \psi_i - \sum_{l \in \delta(k)} \gamma(U, k, l) - \sum_{l \in \delta(k)} \rho(U, k, l) \\
& -\phi_k \leq C_S, \forall k \in V_M, \forall (U, k) \in \mathcal{S}_{*k}. \quad (7) \\
& \omega_{ij}^{kl} \leq d_{ij}^{kl}, \forall ij \in E_I, \forall (k, l) \in \bar{A}_M. \quad (8) \\
& \phi_k \geq 0, \forall k \in V_M. \quad (9) \\
& \varphi_{ij}^{kl} \geq 0, \forall (k, l) \in \bar{A}_M, \forall ij \in E_I. \quad (10) \\
& \omega_{ij}^{kl} \geq 0, \forall (k, l) \in A_M, \forall ij \in E_I. \quad (11)
\end{aligned}$$

Thus, given a variable x_S , where $S = (U, k)$, the reduced price π_S of x_S is calculated as follows: $\pi_S = C_S - \sum_{i \in U} \psi_i + \phi_k + \sum_{l \in \delta(k)} \gamma(U, k, l) + \sum_{l \in \delta(k)} \rho(U, k, l) = \sum_{i \in U} \beta_{ik} + \sum_{ij \in E_I | i, j \in U} d_{ij}^{kk} + \phi_k$, where $\beta_{ik} = c_{ik} - \psi_i + \sum_{j \in \delta(i)} (\sum_{l \in \delta(k)} \varphi_{ij}^{kl} + \sum_{l \in \delta(k)} \omega_{ij}^{kl})$.

Now, we introduce an ILP formulation for the pricing problem for a fixed $k \in V_M$ (i.e. we solve the ILP presented below for each $k \in V_M$ in order to solve the pricing problem). For all $i \in V_I$, let $y_i \in \{0, 1\}$ be such that $y_i = 1$ if and only if $i \in U$. For all $ij \in E_I$, let $h_{ij} \in \mathbb{R}_+$ be such that $h_{ij} = 1$ if and only if $i, j \in U$.

In order to satisfy the connectivity constraint, we include multi-commodity flow constraints (see Gouveia [10]) in our formulation. For all $i \in V_I$, let $w_i \in \{0, 1\}$ be such that $w_i = 1$ if and only if i is the source of the flow. For all $i \in V_I$, let $F_i \in \mathbb{R}_+$ be the units of flow generated in i . Finally, for all $ij \in E_I$, let $f_{ij} \in \mathbb{R}_+$ be the units of flow going from i to j . The main idea of the multi-commodity flow approach is the following: one vertex of U is chosen to be the source of $|U|$ units of flow; the capacity of an edge $ij \in E_I$ is $|V_I|$ if $i, j \in U$, or zero otherwise; each vertex of U must be a sink with demand of one unit of flow; all units of flow must be routed to the sinks. Thus, we have that $G_I[U]$ is connected if and only if there exists a flow satisfying all these conditions.

Below we present an ILP formulation, namely (P_{MRP}), for the pricing problem for a fixed $k \in V_M$. The objective is to find a cluster $S = (U, k)$ such that π_S is minimum. Since (P_{MRP}) is a minimization problem and the coefficients d_{ij} are nonnegative for all $ij \in E_I$, and considering the constraints (12), we have that $h_{ij} = 1$ if and only if $y_i = y_j = 1$.

Constraints (13) guarantee that there is a single source. Constraints (14) guarantee that only the vertices of U are allowed to be the source. Constraints (15) guarantee that if a vertex is not the source then it can

not generate flow. Constraints (16) guarantee that the total flow generated in the source must be equal to $|U|$. Constraints (17) guarantee that each vertex of U is a sink with demand of one unit of flow. Finally, constraints (18) guarantee that a positive amount of flow can be routed from i to j only if $i \in U$ and $ij \in E_I$.

$$\begin{aligned}
\min \quad & \sum_{i \in V_I} \beta_{ik} y_i + \sum_{ij \in E_I} d_{ij}^{kk} h_{ij} + \phi_k \\
\text{s.t.} \quad & y_i + y_j - 1 \leq h_{ij}, \forall ij \in E_I. \quad (12) \\
& \sum_{i \in V_I} w_i = 1. \quad (13) \\
& w_i \leq y_i, \forall i \in V_I. \quad (14) \\
& F_i - |V_I| w_i \leq 0, \forall i \in V_I. \quad (15) \\
& \sum_{i \in V_I} F_i - \sum_{i \in V_I} y_i = 0. \quad (16) \\
& F_i + \sum_{j \in \delta(i)} (f_{ji} - f_{ij}) = y_i, \forall i \in V_I. \quad (17) \\
& \sum_{j \in \delta(i)} f_{ij} - |V_I| y_i \leq 0, \forall i \in V_I. \quad (18) \\
& y_i, w_i \in \{0, 1\}, \quad \forall i \in V_I. \quad (19) \\
& 0 \leq h_{ij}, u_{ij} \leq 1, \quad \forall ij \in E_I. \quad (20) \\
& 0 \leq f_{ij}, f_{ji} \leq |V_I|, \quad \forall ij \in E_I. \quad (21) \\
& 0 \leq F_i \leq |V_I|, \quad \forall i \in V_I. \quad (22)
\end{aligned}$$

In our implementation, instead of solving (P_{MRP}) at each pricing step, in many steps we run only a simple greedy heuristic for building U iteratively using a priority queue, where the minimum increasing on the reduced price of $S(U, k)$ is used as the criterion for selecting the vertices to enter in the set U .

IV. COMPUTATIONAL EXPERIMENTS

First, we describe how the input graphs and the cost functions are generated. We choose parameters τ and λ , and consider the set of characteristics represented by $\Gamma = \{\tau, 2\tau, 3\tau, \dots, \lambda\tau\}$ (in our experiments we choose $\tau = 10$ and $\lambda = 2|V_I|$). All the edges and vertices of both graphs will have a characteristic represented by an element of Γ , and we denote these values as follows: $r_i \in \Gamma$, for all $i \in V_I$; $q_k \in \Gamma$, for all $k \in V_M$; $w_{ij} \in \Gamma$, for all $ij \in E_I$; and $z_{kl} \in \Gamma$, for all $kl \in E_M$. For the vertices $k \in V_M$ we associate another characteristic $h_k \in \Gamma$.

We generate the graph G_I randomly, and then we split the vertices of V_I in t disjoint subsets such that $V_I = V_1 \cup V_2 \cup \dots \cup V_t$ and, for $k = 1, 2, \dots, t$, we have that $V_k \neq \emptyset$ and $G_I[V_k]$ is connected. Define $E_k = \{ij \in E_I \mid i, j \in V_k\}$ and $E_{kl} = \{ij \in E_I \mid i \in V_k \text{ and } j \in V_l\}$. For each set V_k , E_k and E_{kl} , we choose $r_k, w_k, w_{kl} \in \Gamma$ at random and set

Table I
COMPARING HR1 TO HR2

Perturbation		Hr1		Hr2	
κ	η	Time	% CM	Time	% CM
2τ	2τ	711s	100%	178s	100 %
2τ	4τ	595s	100%	186s	100 %
2τ	8τ	709s	100%	181s	100 %
4τ	2τ	942s	100%	176s	100 %
4τ	4τ	785s	100%	167s	100 %
4τ	8τ	901s	100%	169s	100 %
8τ	2τ	846s	100%	163s	90 %
8τ	4τ	891s	100%	168s	90 %
8τ	8τ	2012s	100%	181s	70 %

$r_i = r_k, w_{ij} = w_k, w_{ij} = w_{kl}$, for all $i \in V_k, ij \in E_k$ and $ij \in E_{kl}$, respectively. The next step is to generate G_M . For each set V_k we have a vertex k in V_M such that $q_k = r_k$. For each pair of different vertices $k, l \in V_M$ there is an edge $kl \in E_M$ if and only if there is an edge $ij \in E_I$ such that $i \in V_k$ and $j \in V_l$. For each edge $kl \in E_M$ we set $z_{kl} = w_{kl}$. We then set $h_k = w_k$, for $k = 1, 2, \dots, t$. Finally, we choose two parameters κ and η , and “perturb” G_I as follows. For each vertex $i \in V_I$ we set $r_i = r_i + \epsilon$, for a random ϵ such that $0 \leq \epsilon \leq \kappa$, and we set $w_{ij} = w_{ij} + \epsilon$, for each edge $ij \in E_I$, for a random ϵ such that $0 \leq \epsilon \leq \eta$. The cost functions are defined in the following way: $c_{ik} = |r_i - q_k|$, for all $i \in V_I$ and $k \in V_M$; $d_{ij}^{kl} = |w_{ij} - z_{kl}|$, for all $ij \in E_I$ and $kl \in E_M$; and $d_{ij}^{kk} = |w_{ij} - h_k|$, for all $ij \in E_I$ and $k \in V_M$.

Let α^* be such that $\alpha^*(i) = k$, for all $i \in V_k$, for $k = 1, 2, \dots, t$. Note that α^* is feasible and if we choose $\kappa = 0$ and $\eta = 0$ then $C(\alpha^*) = 0$ (thus, α^* is optimum in this case). In the experiments, we choose different values for κ and η and analyze how far the obtained matching is from α^* .

We used *Gurobi 2.0* © (www.gurobi.com) as the MIP solver and a computer with 8 processors of 2.83GHz and 32GB of RAM. In table I we compare two heuristics, namely Hr1 and Hr2, where Hr1 solves (LP_{GMP}) to optimality and Hr2 does not (instead, Hr2 uses the heuristic described at the end of last section for solving the pricing steps). Each row of column Hr1 corresponds to an instance such that $|V_I| = 200, |E_I| = 600$ and $|V_M| = 30$, and each row of column Hr2 corresponds to the arithmetic average of 10 different instances such that $|V_I| = 500, |E_I| = 1500$ and $|V_M| = 50$. The first two columns indicate the perturbation parameters. Columns “Time” and “% CM” contain the running time and the percentage of correct matchings of vertices, respectively.

All instances in column Hr1 were solved to optimality, i.e. the optimal values of both the LP and the restricted IP are equal, which indicates the robustness

of the method. As shown, the heuristic Hr2 is able to solve larger size instances in less time, comparing to Hr1, but with some loss of the solution’s quality in the instances with large values for κ and η .

V. CONCLUDING REMARKS AND FUTURE WORK

We present a formulation for the graph matching problem and employ the column generation technique to solve instances of the problem. As the experiments show, the method is quite tolerant to “perturbation” on the input graphs. The next step of this research is to apply the method to instances of some applications and compare it with other conventional algorithms.

Acknowledgments: FINEP, FAPESP, CAPES and CNPq (BioCORE project).

REFERENCES

- [1] H. Bunke and G. Allermann, “Inexact graph matching for structural pattern recognition,” *Pattern Recognition Letters*, vol. 1, no. 4, pp. 245 – 253, 1983.
- [2] D. Conte, P. Foggia, C. Sansone, and M. Vento, “Graph matching applications in pattern recognition and image processing,” in *IEEE – ICIP’03*, vol. 2, 2003, pp. 21–24.
- [3] B. Luo and E. Hancock, “Structural graph matching using the em algorithm and singular value decomposition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1120–1136, 2001.
- [4] L. Chen, J. McAuley, R. Feris, T. Caetano, and M. Turk, “Shape classification through structured learning of matching measures,” in *Proc. of CVPR’09*, pp. 365–372.
- [5] P. F. Felzenszwalb and J. D. Schwartz, “Hierarchical matching of deformable shapes,” in *Proc. of CVPR’07*, pp. 1–8.
- [6] A. Moreno, C. Takemura, O. Colliot, O. Camara, and I. Bloch, “Using anatomical knowledge expressed as fuzzy constraints to segment the heart in ct images,” *Pattern Recognition*, vol. 41, no. 8, pp. 2525 – 2540, 2008.
- [7] R. M. Cesar, Jr., E. Bengoetxea, I. Bloch, and P. Larrañaga, “Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms,” *Pattern Recognition*, vol. 38, no. 11, pp. 2099 – 2113, 2005.
- [8] M. C. Boeres, C. C. Ribeiro, and I. Bloch, *Experimental and Efficient Algorithms*, ser. LNCS, April 2004, vol. 3059/2004, ch. A Randomized Heuristic for Scene Recognition by Graph Matching, pp. 100–113.
- [9] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, “Branch-and-price: Column generation for solving huge integer programs,” *Operational Research*, no. 3, pp. 316–329, 1998.
- [10] L. Gouveia, “Multicommodity flow models for spanning trees with hop constraints,” *European Journal of Operational Research*, vol. 95, no. 1, pp. 178–190, 1996.