

A Necessary and Sufficient Condition for Graph Matching to be equivalent to Clique Search

Brijnesh J. Jain and Klaus Obermayer
Berlin University of Technology
Germany

15. Dezember 2009

This paper formulates a necessary and sufficient condition for a generic graph matching problem to be equivalent to the maximum vertex and edge weight clique problem in a derived association graph. The consequences of this results are threefold: first, the condition is general enough to cover a broad range of practical graph matching problems; second, a proof to establish equivalence between graph matching and clique search reduces to showing that a given graph matching problem satisfies the proposed condition; and third, the result sets the scene for generic continuous solutions for a broad range of graph matching problems. To illustrate the mathematical framework, we apply it to a number of graph matching problems, including the problem of determining the graph edit distance.

1 Introduction

The poor representational capabilities of feature vectors triggered the field of *structural pattern recognition* in the early 1970s to focus on pattern analysis tasks, where structured data is represented in terms of strings, trees, or graphs. A key issue in structural pattern recognition is to measure the proximity of two structural descriptions in terms of their structurally consistent or inconsistent parts. There is also strong evidence that human cognitive models of comparison and analogy establish relational correspondences between structured objects [8, 11, 9]. The problem of measuring the structural proximity of graphs, more generally referred to as the *graph matching problem*, is often computationally inefficient. Therefore, an appropriate formulation of the graph matching problem is important to devise efficient solutions and to gain insight into the nature of the problem.

One popular technique is to transform graph matching to an equivalent clique search in a derived auxiliary structure, called *association graph* [1, 2, 18, 15, 19]. Chen & Yun [6]

generalized association graph techniques by compiling results from [2, 5, 12] and showing that the maximum common (induced) subgraph problem and its derivations can be casted to a maximum clique problem. Pelillo [16] extended this collection by transforming the problem of matching free trees¹ to the MVCP so that connectivity is preserved. Bunke [4] showed that for special cost functions, the error correcting graph matching problem and the maximum common subgraph are equivalent. As a consequence, special graph edit distances can be computed via clique search in an association graph. Schädler & Wysotzki [20] mapped the best monomorphic graph matching problem to a maximum weighted clique problem without presenting a sound theoretical justification of their transformation.

Although association graph techniques are attractive, an equivalence relationship between graph matching and clique search has been established only for certain classes of graph matching problems. Examples for graph matching problems, which have been proven equivalent to clique search in an association graph are the maximum common (induced) subgraph and special cases thereof, or the graph and tree edit distance for special cost functions. To utilize the benefits of association graph techniques for a possibly broad range of graph matching problems, we pose the following question:

Under which conditions is graph matching equivalent to clique search in an association graph?

With the above question in mind, we make the following contributions:

- We formulate a necessary and sufficient condition (C), for graph matching problems to be equivalent to the maximum weight clique problem in an association graph. As opposed to standard formulations, the maximum weight clique problem takes into account weights assigned to both vertices and edges.
- Condition (C) is sufficiently strict to cover a broad range of graph matching problems. Besides the well-known standard problems, we show, for example, that determining the graph edit distance is equivalent to the maximum weight clique problem in an association graph.
- The mathematical framework developed in this paper provides a proof technique that reduces the problem of showing equivalence between graph matching and clique search to the problem of showing that a given graph matching problem satisfies condition (C). By means of a number of examples, we illustrate how the novel technique considerably simplifies equivalence proofs.

The paper is organized as follows. We conclude this section by introducing the terminology. Section 2 presents the background and an intuitive idea of condition (C). In Section 3, we formulate condition (C) and prove that (C) is necessary and sufficient for the desired equivalence relationship. We apply (C) in Section 4. Finally, Section 5 concludes with a summary of the main results and an outlook on further research.

¹A free tree is a directed acyclic graph without a *root*.

1.1 Preliminaries

The aim of this subsection is to introduce the terminology and notations used throughout this contribution. We assume knowledge about basic graph theory.

Basic Graph Theory

Graphs For convenience of presentation, all graphs are undirected without loops. The graphs we consider are triples $X = (V, E, \mathbf{X})$ consisting of a finite set $V = V(X)$ of vertices, a set $E = E(X)$ of edges, and an *attributed adjacency matrix* $\mathbf{X} = (x_{ij})$ with elements x_{ij} from a set of *attributes* $\mathcal{A} \cup \{\epsilon\}$. Elements $x_{ij} \in \mathcal{A}$ are attributes either assigned to vertices if $i = j$ or to edges if $(i, j) \in E$. We label non-edges $(i, j) \in \bar{E}$ with a distinguished *void attribute* $x_{ij} = \epsilon$.

Subgraphs We write $X' \subseteq X$ to denote that X' is a subgraph of X . For a subset U of $V(X)$, the graph $X[U]$ denotes the induced subgraph of X induced by U .

Items To unclutter the text from tedious case distinctions, we occasionally make use of the notion of *item*. Items of X are elements from $I(X) = V \times V$. Thus, an item $\mathbf{i} = (i, j)$ of X is either a vertex if $i = j$, an edge if $\mathbf{i} \in E(X)$, or a non-edge if $\mathbf{i} \in \bar{E}(X)$.

Morphisms Let X and Y be graphs. A *morphism* from X to Y is a mapping

$$\phi : V(X) \rightarrow V(Y), \quad i \mapsto i^\phi.$$

A *partial morphism* from X to Y is a morphism ϕ defined on a subset of $V(X)$. By $\mathcal{D}(\phi)$ we denote the *domain* and by $\mathcal{R}(\phi)$ the *range* of a partial morphism ϕ . By abuse of notation we occasionally write $\mathbf{i} = (i, j) \in \mathcal{D}(\phi)$ if $i \in \mathcal{D}(\phi)$ and $j \in \mathcal{D}(\phi)$. The meaning of $\mathbf{i} \in \mathcal{R}(\phi)$ is now obvious. A *monomorphism* is an injective morphism. We use the notions *homo-* and *isomorphism* as in standard graph theory. A partial *subgraph-morphism* is an isomorphism between subgraphs.

The Maximum Weight Clique Problem

Let $X = (V, E, \mathbf{X})$ be a graph with attributes from $\mathcal{A} = \mathbb{R} \cup \{\epsilon\}$. A *clique* of X is a subset $C \subseteq V$ such that the induced subgraph $X[C]$ is complete. A clique is said to be *maximal* if C is not contained in any larger clique of X , and *maximum* if C has maximum cardinality of vertices. The *weight* $\omega(C)$ of a clique C of X is defined by

$$\omega(C) = \sum_{i,j \in C} x_{ij}. \tag{1}$$

The weight of a clique C is the total of all vertex and edge weights of the subgraph $X[C]$ induced by C . Since the vertices of $X[C]$ are mutually adjacent, the void symbol ϵ does not occur in the sum of (1). Hence, $\omega(C)$ is well-defined.

The *maximum weight clique problem* is a combinatorial optimization problem of the form

$$\begin{aligned} & \text{maximize} && \omega(C) = \sum_{i,j \in C} x_{ij} \\ & \text{subject to} && C \in \mathcal{C}_X, \end{aligned} \tag{2}$$

where \mathcal{C}_X is the set of all cliques of X .

A *maximal weight clique* of X is a clique $C \in \mathcal{C}_X$ such that

$$C \subseteq C' \Rightarrow \omega(C) \geq \omega(C')$$

for all cliques C' of X . It is impossible to enlarge a maximal weight clique C to a clique with higher weight. If all vertices and edges of X are associated with positive weights, a maximal weight clique is not a proper subset of another clique. A *maximum weight clique* of X is a clique $C \in \mathcal{C}_X$ with maximum total weight over its vertices and edges. By \mathcal{C}_X^\times we denote the set of all maximal cliques and by \mathcal{C}_X^* the set of all maximum cliques of X .

2 Background

The aim of this section is twofold: First, it introduces the problem and motivates its solution. To this end, we consider the classical maximum common induced subgraph problem (MCISP). Second, it provides an intuitive idea of how to solve that problem in a more general setting.

2.1 The Problem

To set the scene, we consider the MCISP. Given two graphs X and Y , the *MCISP* asks for a partial isomorphism $\phi : V(X) \rightarrow V(Y)$ that maximizes the cardinality $|\mathcal{D}(\phi)|$ of its domain. This optimization problem is called MCISP, because each partial isomorphism ϕ between X and Y is an isomorphism between the induced subgraphs $X[\mathcal{D}(\phi)]$ and $Y[\mathcal{R}(\phi)]$. Since maximizing $|\mathcal{D}(\phi)|^2$ instead of $|\mathcal{D}(\phi)|$ does not effect the problem, we may rewrite the MCISP as

$$\begin{aligned} & \text{maximize} && f(\phi, X, Y) = |\mathcal{D}(\phi)|^2 = \sum_{i,j \in \mathcal{D}(\phi)} \kappa_{ij}^{\phi} \\ & \text{subject to} && \phi \in \mathcal{M}, \end{aligned} \tag{3}$$

where the *search space* \mathcal{M} is the set of all partial isomorphisms from X to Y . The values $\kappa_{ijrs} \in \{0, 1\}$ are the *compatibility values* with

$$\kappa_{ijrs} = \begin{cases} 1 & : x_{ij} = y_{rs} \\ 0 & : \text{otherwise} \end{cases} \tag{4}$$

for all $i, j \in V(X)$ and $r, s \in V(Y)$. A compatibility value κ_{ijrs} indicates whether item (i, j) of X has the same attribute as item (r, s) of Y . The *matching objective* f counts the number of items from X consistently mapped to their exact counterparts in Y .

It is well known that the MCISP is NP-complete [7]. Therefore, exact algorithms that guarantee to return an optimal solution are useless for all but the smallest graphs. In a practical setting, the time required to compute an optimal solution will typically reduce the overall *utility* of the algorithm. It is rather more desirable to trade quality with time and to provide near-optimal solutions within an acceptable time limit. Thus, it is conducive to consider local optimal solutions of (3). We say a partial isomorphism $\phi \in \mathcal{M}$ is a local optimal solution if there is no other partial isomorphism from \mathcal{M} with larger domain.

To solve the maximum common induced subgraph problem, we transform it to another combinatorial optimization problem using association graph techniques originally introduced by [1, 14, 2]. An association graph $Z = X \diamond Y$ of X and Y is a graph with vertex and edge set

$$\begin{aligned} V(Z) &= \{ir : x_{ii} = y_{rr}\} \subseteq V(X) \times V(Y) \\ E(Z) &= \{(ir, js) : x_{ij} = y_{rs}\} \subseteq V(Z) \times V(Z). \end{aligned}$$

The attributed adjacency matrix $\mathbf{Z} = (z_{irjs})$ of Z is defined by

$$z_{irjs} = \begin{cases} 1 & : \quad ir = js \text{ and } ir \in V(Z) \\ 1 & : \quad ir \neq js \text{ and } (ij, rs) \in E(Z) \\ \epsilon & : \quad \text{otherwise} \end{cases}$$

for all $ir, js \in V(Z)$. Thus, we assign to all vertices and edges of Z the weight 1 and to all non-edges the void attribute ϵ . Note that we derived the weights z_{irjs} by inserting the corresponding compatibility values κ_{ijrs} .

By definition of an association graph, we have the following useful equivalence relationship between the partial isomorphisms from \mathcal{M} and the cliques in Z :

- Z uniquely encodes each partial isomorphism ϕ from \mathcal{M} as a clique C_ϕ in Z such that $|\mathcal{D}(\phi)| = |C|$.
- The set $\mathcal{C}_Z(\mathcal{M})$ of cliques encoding partial isomorphisms is equal to the set \mathcal{C}_Z of all cliques in Z .

The above equivalence directly implies that the maximum (maximal) cliques of Z are in one-to-one correspondence with the global (local) optimal partial isomorphisms from \mathcal{M} . Hence, solving the MCISP (3) is equivalent to solving the maximum clique problem in Z .

Benefits of the Association Graph Framework

What makes an association graph formulation of the MCISP so useful is that

- the maximum clique problem is mathematically well founded,
- it provides us access to a plethora of clique algorithms to solve the original problem,
- abstracts from the particularities of the graphs being matched, and

- abstracts from the constraints on feasible matches.

The benefits of the second item on clique algorithms is worth to be discussed in more detail. Both the MCISP and the maximum clique problem are combinatorial problems. Solution techniques for combinatorial problems can be classified into two groups:

- *Discrete methods* generate a sequence of suboptimal or partial solutions to the original problem. This process is guided by local search combined with techniques to escape from local optimal solutions.
- *Continuous methods* embed the discrete solution space in a larger continuous one. Exploiting the topological and geometric properties of the continuous space, the algorithm creates a sequence of points that converges to the solution of the original problem.

Discrete approaches suffer from problems having a large number of local maxima. On the other hand, continuous approaches such as interior point methods reveal to be efficient solutions to large scale combinatorial optimization problems. Hence, using association graph techniques provides us access to those efficient continuous solutions.

Shortcomings of the Association Graph Framework

Despite its long tradition and benefits, the association graph formulation still suffers from the following shortcomings:

- It is unclear which graph matching problems are equivalent to clique search in an association graph. The equivalence relationship in question has been proved only for selected problems. For example, Chen & Yun [6] compiled results from [2, 5, 12], showing that the maximum common (induced) subgraph problem and its derivations can be casted to a maximum clique problem. Pelillo and his co-workers extended this collection for different types of tree matching problems [3, 18, 17, 16].
- In all these examples, equivalence proofs of graph matching and clique search follow the same recurring pattern: First construct an appropriate association graph and then establish a bijective mapping between cliques and morphisms. There is no mathematical framework that reduces equivalence proofs to showing that the preconditions of some generic equivalence relationship between graph matching and clique search are satisfied.

What we want

The aim is to remove both shortcomings addressed in the previous paragraph. We want to formulate a necessary and sufficient condition (C) for equivalence between graph matching and clique search. The existence of condition (C) explains which graph matching problems are equivalent to clique search in an association graph and reduces equivalence proofs to showing that a given graph matching problem satisfies (C). In addition, we want to indicate by a number of examples that a broad range of graph matching problems satisfy condition (C) and are therefore equivalent to clique search in an association graph.

2.2 The Idea

Our goal is to present an intuitive idea of a necessary and sufficient condition (C) for equivalence between graph matching and clique search. To this end, we turn from the MCISP to the general case. Suppose that X and Y are two graphs, and let \mathcal{M}_{XY} be the set of all partial morphisms from X to Y . The graph matching problem considered here generalizes the MCISP in two ways:

- By allowing nonnegative real-valued compatibility values $\kappa_{ijrs} = \kappa_{jisr}$.
- By allowing any subset of \mathcal{M}_{XY} as search space.

Allowing real-valued compatibility values $\kappa_{ijrs} \geq 0$ enables us to measure the degree of *compatibility*, or *consistency*, between items $\mathbf{i} = (i, j)$ of X and $\mathbf{r} = (r, s)$ of Y . This generalization is useful to cope with noisy attributes. Considering arbitrary subsets of \mathcal{M}_{XY} provides a flexible mechanism to cope with structural errors.

Applying both generalizations, the *graph matching problem* (GMP) is a combinatorial optimization problem of the form

$$\begin{aligned} \text{maximize} \quad & f(\phi, X, Y) = \sum_{\mathbf{i} \in \mathcal{D}(\phi)} \kappa_{\mathbf{i}\mathbf{i}\phi} \\ \text{subject to} \quad & \phi \in \mathcal{M}, \end{aligned} \tag{5}$$

where the search space \mathcal{M} is a subset of \mathcal{M}_{XY} , and $\mathbf{i}^\phi = (i^\phi, j^\phi) \in I(Y)$ is the image item of item $\mathbf{i} = (i, j) \in I(X)$. Problem (5) is a generic formulation that subsumes a broad range of practical graph matching problems.

Often, solving an instance of problem (5) is computationally intractable. As for the MCISP, we therefore consider simpler versions of (5) that allow local optimal solutions. A partial morphism $\phi \in \mathcal{M}$ is a local optimal solution of (5) if $\mathcal{D}(\phi) \subseteq \mathcal{D}(\psi)$ implies $f(\phi, X, Y) \geq f(\psi, X, Y)$ for all $\psi \in \mathcal{M}$.

Applying the Association Graph Framework

The central question at issue is: Under which conditions is the generic graph matching problem (5) equivalent to clique search in a derived association graph $Z = X \diamond Y$? Certainly, an association graph Z should satisfy the following properties:

- P1.** Z uniquely encodes each partial morphism ϕ from the search space \mathcal{M} as a clique C_ϕ in Z such that $f(\phi, X, Y) = \omega(C)$, where $\omega(C)$ denotes the weight of clique C .
- P2.** The set $\mathcal{C}_Z(\mathcal{M}) = \{C_\phi \in \mathcal{C}_Z : \phi \in \mathcal{M}\}$ of all cliques encoding partial morphisms from \mathcal{M} is equal to the set \mathcal{C}_Z of all cliques in Z .

Ad P1: We can always derive an association graph satisfying the first property P1. Consider the complete graph Z_{XY} with vertex set $V_{XY} = V(X) \times V(Y)$. To each item $((i, j), (r, s)) \in I(Z_{XY})$, we assign the weight κ_{ijrs} . Since Z_{XY} is complete, there are only vertex and edge items. We extract a subgraph from Z_{XY} to form an association graph

Z as follows: For each partial morphism $\phi \in \mathcal{M}$, we construct a complete subgraph Z_ϕ of Z_{XY} with vertex set $V(Z_\phi) = C_\phi = \{ii^\phi : i \in \mathcal{D}(\phi)\}$. The vertex set C_ϕ is a clique in Z_{XY} such that $f(\phi, X, Y) = \omega(V_\phi)$. We obtain an association graph $Z \subseteq Z_{XY}$ by taking the union of all subgraphs Z_ϕ with $\phi \in \mathcal{M}$. Then the construction Z satisfies property P1. Note that construction of Z in the above way is impractical, because it requires enumeration of all members of \mathcal{M} . But it provides a simple way to show that there is always an association graph satisfying property P1.

Ad P2: Now let us turn to the second property P2. We first provide a fictitious example to show that P2 does generally not hold.

Example 1. Consider the set \mathcal{M}_2 of partial morphisms $\phi : V(X) \rightarrow V(Y)$ that are defined on subsets of exactly two vertices from X , i.e. $|\mathcal{D}(\phi)| = 2 < |X|$. Suppose that we construct Z as described in the previous paragraph. Then the set $\mathcal{C}_Z(\mathcal{M}_2)$ of all cliques encoding morphisms from \mathcal{M}_2 are cliques in Z with exactly two vertices. We may encounter the following pitfalls:

PF1. Let $C = \{i, j\}$ be a clique in Z encoding $\phi \in \mathcal{M}_2$. Then $\{i\}$ and $\{j\}$ are both cliques in Z not contained in $\mathcal{C}_Z(\mathcal{M}_2)$.

PF2. Let C, C', C'' be cliques from $\mathcal{C}_Z(\mathcal{M}_2)$ with $C = \{i, j\}$, $C' = \{j, k\}$, and $C'' = \{i, k\}$. Then $C = \{i, j, k\}$ is a clique in Z with three vertices and therefore not contained in $\mathcal{C}_Z(\mathcal{M}_2)$.

Hence, we have $\mathcal{C}_Z(\mathcal{M}_2) \neq \mathcal{C}_Z$ and therefore a graph matching problem (5) defined on \mathcal{M}_2 is not equivalent to clique search in an association graph.

Example 1 indicates that equivalence between graph matching and clique search depends on the structure of the search space \mathcal{M} . So our central question of issue reduces to a necessary and sufficient condition on \mathcal{M} such that $\mathcal{C}_Z(\mathcal{M}) = \mathcal{C}_Z$.

A Necessary and Sufficient Condition

The goal is to present an intuitive idea of a necessary and sufficient condition for $\mathcal{C}_Z(\mathcal{M}) = \mathcal{C}_Z$. Our claim is that $\mathcal{C}_Z(\mathcal{M}) = \mathcal{C}_Z$ holds whenever there is a *property* that completely describes the set \mathcal{M} .

First, we specify the notion of property. Given a graph matching problem, the purpose of a property is to describe the characteristics of the search space \mathcal{M} in such a way that we can derive the desired equivalence relationship. For example, let \mathcal{M} be the set of all partial morphisms ϕ from \mathcal{M}_{XY} that satisfy the property \mathfrak{p} to be injective. Although property \mathfrak{p} completely describes the characteristics of \mathcal{M} , it is not suitable to deal with it conveniently. The reason is as follows: According to the pitfalls PF1 and PF2 of Example 1, we have to show that restrictions and feasible unions of morphisms that satisfy property \mathfrak{p} also satisfy \mathfrak{p} . This task is unnecessarily complex and can be simplified by imposing a locality restriction on the notion of property.

A *property* \mathfrak{p} on a set \mathcal{S} is a binary function $\mathfrak{p} : \mathcal{S} \rightarrow \{0, 1\}$. We say an element $x \in \mathcal{S}$ satisfies property \mathfrak{p} if $\mathfrak{p}(x) = 1$ for each $x \in \mathcal{S}'$. We can extend the notion of a property

to the subsets of \mathcal{S} by considering their local behavior. A subset $\mathcal{S}' \subseteq \mathcal{S}$ satisfies property \mathfrak{p} if $\mathfrak{p}(x) = 1$ for each element $x \in \mathcal{S}'$, i.e. if \mathcal{S}' locally satisfies \mathfrak{p} .

Next, we characterize an association graph $Z = X \diamond Y$ by means of a property. To this end, we consider the complete graph Z_{XY} defined on the vertex set $V_{XY} = V(X) \times V(Y)$. Let \mathfrak{p} be a property on the set $I_{XY} = I(Z_{XY})$ of items such that the vertices and edges of Z are all the items that satisfy \mathfrak{p} . Hence, the subsets of I_{XY} that satisfy property \mathfrak{p} are the induced subgraphs of Z induced by its cliques. We can express the set of all cliques in Z as $\mathcal{C}_Z = \{U \subseteq I_{XY} : U \text{ satisfies } \mathfrak{p}\}$. Now using \mathfrak{p} , we rewrite condition $\mathcal{C}_Z(\mathcal{M}) = \mathcal{C}_Z$ as follows: *There is a property \mathfrak{p} such that $\mathcal{C}_Z(\mathcal{M}) = \{C \in \mathcal{C}_Z : C \text{ satisfies } \mathfrak{p}\}$.* Since the cliques from $\mathcal{C}_Z(\mathcal{M})$ encode the feasible morphisms from \mathcal{M} , we may directly translate the last condition to a condition on the search space \mathcal{M} :

(C) *There is a property \mathfrak{p} such that $\mathcal{M} = \{\phi \in \mathcal{M}_{XY} : \phi \text{ satisfies } \mathfrak{p}\}$.*

To summarize, we claim that condition (C) is necessary and sufficient for the equivalence relationship in question.

3 A Necessary and Sufficient Condition

The aim of this section is to prove that condition (C) is necessary and sufficient for equivalence between graph matching and clique search in an association graph.

3.1 Closed Sets of \mathfrak{p} -Morphisms

Here, we consider properties \mathfrak{p} defined on the set $I_{XY} = I(X) \times I(Y)$ of pairs of items from X and Y . Let $(\mathbf{i}, \mathbf{r}) \in I_{XY}$. We say, items \mathbf{i} and \mathbf{r} are \mathfrak{p} -similar, written as $\mathbf{i} \sim_{\mathfrak{p}} \mathbf{r}$, if (\mathbf{i}, \mathbf{r}) satisfies property \mathfrak{p} . Let us consider some examples.

Example 2. *Let $(\mathbf{i}, \mathbf{r}) \in I_{XY}$, and let all morphisms ϕ be partial morphisms from \mathcal{M}_{XY} . Then the following examples are properties on I_{XY} :*

1. $\mathbf{i} \sim_{\mathfrak{p}} \mathbf{r} =$ *there is a partial (mono-, subgraph-, homo-, iso-) morphism ϕ with $\mathbf{i}^\phi = \mathbf{j}$.*
2. $\mathbf{i} \sim_{\mathfrak{p}} \mathbf{r} =$ *there is a partial connectivity preserving isomorphism ϕ with $\mathbf{i}^\phi = \mathbf{j}$.*

Next, we apply the notion of property on $I_{XY} = I(X) \times I(Y)$ to partial morphisms. To this end, we identify a partial morphism $\phi \in \mathcal{M}_{XY}$ with the binary relation

$$\Gamma(\phi) = \{(\mathbf{i}, \mathbf{r}) \in \mathcal{D}(\phi) \times \mathcal{R}(\phi) : \mathbf{i}^\phi = \mathbf{r}\} \subseteq I_{XY}.$$

We say, ϕ is a \mathfrak{p} -morphism if $\Gamma(\phi)$ is a subset of the set $\mathcal{R}_{XY}^{\mathfrak{p}} = \{(\mathbf{i}, \mathbf{r}) \in I_{XY} : \mathbf{i} \sim_{\mathfrak{p}} \mathbf{r}\}$. In informal terms, a partial morphism ϕ satisfies property \mathfrak{p} if it locally satisfies \mathfrak{p} . To illustrate the definition of \mathfrak{p} -morphism, we provide some examples.

Example 3. *Consider the properties \mathfrak{p} of Example 2.*

1. Let \mathfrak{p} be the property of Example 2.1, and let $\phi \in \mathcal{M}_{XY}$ be a partial morphism. Since $\mathbf{i}^\phi = \mathbf{r}$ for all pairs $(\mathbf{i}, \mathbf{r}) \in \Gamma(\phi)$, we trivially have $\mathbf{i} \sim_{\mathfrak{p}} \mathbf{r}$ for all pairs $(\mathbf{i}, \mathbf{r}) \in \Gamma(\phi)$. Hence, $\Gamma(\phi) \subseteq \mathcal{R}_{XY}^{\mathfrak{p}}$ and therefore ϕ is a \mathfrak{p} -morphism. Similarly, mono-, homo-, and isomorphisms are \mathfrak{p} -morphisms.
2. Let \mathfrak{p} be the property of Example 2.2, and let ϕ be a partial connectivity preserving morphism on $V(X)$. As in (1), we find that ϕ is a \mathfrak{p} -morphism. Note that for non-edge items $(i, j) \in \overline{E}(X)$ the restriction ϕ' of ϕ to $\{i, j\}$ does not preserve connectivity. But what is important is that ϕ' can be extended to a partial morphism ϕ that preserves connectivity.

Let $\mathcal{M}_{XY}^{\mathfrak{p}} \subseteq \mathcal{M}_{XY}$ denote the set of all \mathfrak{p} -morphisms. We say, a subset \mathcal{M} of \mathcal{M}_{XY} is \mathfrak{p} -closed if $\mathcal{M} = \mathcal{M}_{XY}^{\mathfrak{p}}$. The next result provides basic \mathfrak{p} -closed sets.

Proposition 1. *The following subsets of \mathcal{M}_{XY} are \mathfrak{p} -closed:*

1. $\mathcal{M} =$ set of all partial morphisms
2. $\mathcal{M} =$ set of all partial monomorphisms
3. $\mathcal{M} =$ set of all partial homomorphisms
4. $\mathcal{M} =$ set of all partial isomorphisms
5. $\mathcal{M} =$ set of all partial subgraph-morphisms

Proof: We only show the assertion for the set \mathcal{M} of all partial isomorphisms. The proofs for the other sets are similar. Let X and Y be graphs with adjacency matrices $\mathbf{X} = (x_{ij})$ and $\mathbf{Y} = (y_{rs})$. We define a property \mathfrak{p} on I_{XY} such that

$$\mathcal{R}_{XY}^{\mathfrak{p}} = \{(\mathbf{i}, \mathbf{r}) : x_{\mathbf{i}} = y_{\mathbf{r}}, \text{type}(\mathbf{i}) = \text{type}(\mathbf{r})\} \subseteq I(X) \times I(Y),$$

where $\text{type}(\mathbf{i})$ maps item \mathbf{i} to its type (vertex, edge, or non-edge).

Next, we show that $\mathcal{M} = \mathcal{M}_{XY}^{\mathfrak{p}}$. Let ϕ be a partial isomorphism from X to Y , let \mathbf{i} be an item of X , and let $\mathbf{i}^\phi = \mathbf{r}$ be the image of \mathbf{i} in Y . Since ϕ is a partial isomorphism, we have $x_{\mathbf{i}} = y_{\mathbf{r}}$. In addition, ϕ preserves the type of items \mathbf{i} and \mathbf{r} . Hence, the set $\Gamma(\phi)$ is a subset of $\mathcal{R}_{XY}^{\mathfrak{p}}$. This proves $\mathcal{M} \subseteq \mathcal{M}_{XY}^{\mathfrak{p}}$.

Now assume that ϕ is a \mathfrak{p} -morphism from $\mathcal{M}_{XY}^{\mathfrak{p}}$. Since $\Gamma(\phi)$ is a subset of $\mathcal{R}_{XY}^{\mathfrak{p}}$, it is sufficient to show that ϕ is bijective. Assume that ϕ is not bijective. Then ϕ is not injective, because ϕ is a partial morphism. Hence, there are distinct vertices i, j of X with $i^\phi = j^\phi = r$. Then ϕ maps a non-vertex item (i, j) to a vertex item (r, r) , which contradicts the condition $\text{type}(\mathbf{i}) = \text{type}(\mathbf{r})$. This proves $\mathcal{M}_{XY}^{\mathfrak{p}} \subseteq \mathcal{M}$. Combining both results yields the assertion. \square

Though the proof of Proposition 1 is fairly simple, it illustrates the basic approach how to show that a certain subset \mathcal{M} of \mathcal{M}_{XY} is \mathfrak{p} -closed. The task is to construct an appropriate property \mathfrak{p} such that we can show $\mathcal{M} = \mathcal{M}_{XY}^{\mathfrak{p}}$.

We conclude this section with a formal proof that the set \mathcal{M}_2 from Example 4 is not \mathfrak{p} -closed.

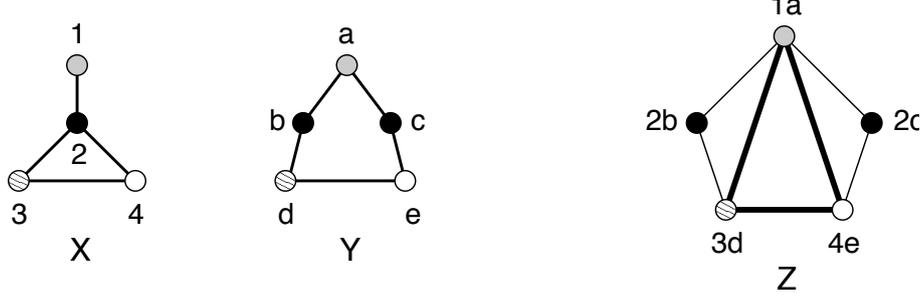


Abbildung 1: Graphs X and Y with association graph $Z = X \diamond Y$ for the maximum common connectivity preserving induced subgraph problem. Different fillings refer to different vertex attributes. All edges have the same attribute. Highlighted edges in X and Y correspond to isomorphic subgraphs. Note that the highlighted subgraph in X is not induced. Highlighted edges in Z refer to the maximum clique.

Example 4. Consider the set \mathcal{M}_m of partial morphisms $\phi : V(X) \rightarrow V(Y)$ with $|\mathcal{D}(\phi)| \leq m < |X|$, where $m \geq 2$. There is no property \mathfrak{p} on I_{XY} such that \mathcal{M}_m is \mathfrak{p} -closed.

Proof: Let \mathfrak{p} denote the property on I_{XY} such that the set of all partial morphisms \mathcal{M}_{XY} is \mathfrak{p} -closed. According to Proposition 1, such a property exists. Consider the set \mathcal{M}_2 . It is easy to see that

$$\bigcup_{\phi \in \mathcal{M}_2} \Gamma(\phi) = \mathcal{R}_{XY}^{\mathfrak{p}}.$$

On one hand, the relation $\mathcal{R}_{XY}^{\mathfrak{p}}$ is too large, because it admits arbitrary partial morphisms as \mathfrak{p} -morphisms. On the other hand, $\mathcal{R}_{XY}^{\mathfrak{p}}$ is a minimal set in the following sense: If we remove an element (i, r) from $\mathcal{R}_{XY}^{\mathfrak{p}}$, then the morphism $\phi \in \mathcal{M}_2$ with $i^\phi = r$ is no longer a \mathfrak{p} -morphism. This shows the assertion. \square

3.2 Construction of an Association Graph

We present the usual constructive definition of an association graph for GMP (5) defined on a \mathfrak{p} -closed search space \mathcal{M} .

For each element $(i, r) \in I_{XY}$, we check whether item i from X and item r from Y can be associated to a vertex or edge in an association graph. The morphisms from the search space \mathcal{M} determine the association rule via its describing property \mathfrak{p} . We associate item i from X with item r from Y if they are \mathfrak{p} -similar. Thus, an *association graph* $Z = X \diamond Y$ is a weighted graph with vertex and edge set

$$V(Z) = \{ir : (i, i) \sim_{\mathfrak{p}} (r, r)\} \subseteq V(X) \times V(Y)$$

$$E(Z) = \{(ir, js) : (i, j) \sim_{\mathfrak{p}} (r, s)\} \subseteq V(Z) \times V(Z).$$

The matrix $Z = (z_{irjs})$ with

$$z_{irjs} = \begin{cases} \epsilon & : (ij, rs) \in \bar{E}(Z) \\ \kappa_{ijrs} & : \text{otherwise} \end{cases}$$

assigns weights to the vertices and edges of Z . Note that an association graph assigns real-valued weights to both, its vertices *and* edges. This is in contrast to standard association graph formulations, where Z is either unweighted (constant weights) or weights are assigned to vertices only.

3.3 A Necessary and Sufficient Condition

In this subsection, we show that \mathfrak{p} -closure of \mathcal{M} is a necessary and sufficient condition for the desired equivalence between graph matching and clique search in an association graph. Note that equivalence means that there is a bijective mapping $\Phi : \mathcal{C}_Z \rightarrow \mathcal{M}$ with $\omega(C) = f(\Phi(C))$ for all cliques $C \in \mathcal{C}_Z$. The mapping Φ then induces a one-to-one correspondence between the maximum (maximal) cliques in Z and the global (local) optimal solutions from \mathcal{M} .

Theorem 1. *Let X and Y be graphs. Then the GMP (5) of X and Y is equivalent to the MWCP in a κ -association graph $Z = X \diamond Y$ if, and only if, there is a property \mathfrak{p} on I_{XY} such that the search space \mathcal{M} of (5) is the set of all \mathfrak{p} -morphisms from \mathcal{M}_{XY} .*

Proof: The \Rightarrow - direction is trivial. Suppose that GMP (5) is equivalent to the MWCP in Z . Then we simply define the property \mathfrak{p} with $i \sim_{\mathfrak{p}} r$ if there is a morphism $\phi \in \mathcal{M}$ with $i^\phi = r$.

Now let us show the opposite direction. Suppose that there is a property \mathfrak{p} on I_{XY} such that \mathcal{M} is the set of all \mathfrak{p} -morphisms from \mathcal{M}_{XY} . We want to show that there is a bijection

$$\Phi : \mathcal{C}_Z \rightarrow \mathcal{M}, \quad C \mapsto \phi_C$$

such that $\omega(C) = f(\phi_C, X, Y)$ for all $C \in \mathcal{C}_Z$. With each clique $C \in \mathcal{C}_Z$ we associate a partial morphism $\phi_C : V(X) \rightarrow V(Y)$ such that $\phi_C(i) = r$ for all $(i, r) \in C$. We show that ϕ_C is a feasible morphism from \mathcal{M} . Let $i, j \in V(X)$ and $r, s \in V(Y)$ be vertices with $\phi_C(i) = r$ and $\phi_C(j) = s$. By construction, (i, r) and (j, s) are members of clique C . Since $Z[C]$ is complete, there is an edge incident with (i, r) and (j, s) . Hence, $(i, r) \sim_{\mathfrak{p}} (j, s)$ and, therefore, $\Gamma(\phi_C) \subseteq \mathcal{R}_{XY}^{\mathfrak{p}}$. Since \mathcal{M} is \mathfrak{p} -closed, we have $\phi_C \in \mathcal{M}$.

Similarly, with each morphism $\phi \in \mathcal{M}$ we associate a subset C_ϕ of $V(Z)$ with

$$(i, j) \in C_\phi \Leftrightarrow \phi(i) = j \text{ or } \phi(j) = i.$$

Since ϕ is a feasible \mathfrak{p} -morphism, C_ϕ is a clique in Z . It is straightforward to show that both associations give rise to well-defined mappings

$$\begin{aligned} \Phi : \mathcal{C}_Z &\rightarrow \mathcal{M}, & C &\mapsto \phi_C \\ \Psi : \mathcal{M} &\rightarrow \mathcal{C}_Z, & \phi &\mapsto C_\phi. \end{aligned}$$

Tabelle 1: Examples of standard graph matching problems.

Type	Graph Matching Problem	$\mathcal{M} \subseteq \mathcal{M}_{XY}$
1	Maximum Common Subgraph Problem → Special case: Subgraph Isomorphism Problem	partial subgraph-morphisms total subgraph-morphisms
2	Maximum Common Induced Subgraph Problem → Special case: Induced Subgraph Isomorphism Problem → Special case: Graph Isomorphism Problem	partial isomorphisms total isomorphisms to subgraph of total isomorphisms
3	Maximum Common Homomorphic Subgraph Problem → Special case: Subgraph Homomorphism Problem	partial homomorphisms total homomorphisms to subgraph

From $\Phi \circ \Psi = \text{id}$ on \mathcal{M} and $\Psi \circ \Phi = \text{id}$ on \mathcal{C}_Z , it follows that Φ is bijective. Finally, the assertion follows from

$$\omega(C) = \sum_{ir, js \in C} z_{irjs} = \sum_{i \in \mathcal{D}(\phi_C)} \kappa_{ii\phi} = f(\phi_C, X, Y).$$

□

3.4 Implications of Theorem 1

Theorem 1 has the following implications:

- Equivalence proofs now reduce to showing that the search space \mathcal{M} of a GMP is \mathfrak{p} -closed for some property \mathfrak{p} .
- Theorem 1 is the starting point for generic continuous solutions to GMPs defined on \mathfrak{p} -closed search spaces. An equivalent continuous formulation of the MWCP is given in [10].

4 Application of Theorem 1

The aim of this section is threefold: First, we show that common graph matching problems satisfy the necessary and sufficient condition. Second, we want to illustrate how the necessary and sufficient condition simplifies equivalence proofs. Third, we show the equivalence relationship for further examples, such as the graph edit distance, for which the desired relationship has been unproven.

4.1 Simple Exact Graph Matching Problems

In this subsection, we consider the problems listed in Table 4.1. For the first two types of problems equivalence to clique search is well-known. Homomorphic GMP of type 3 in Table 4.1 have been considered in a slight variation for attributed trees [3].

An *exact graph matching problem* is a GMP (5), where the compatibility values of the matching objective are of the general form

$$\kappa_{ij} = \begin{cases} \alpha_V & : \mathbf{x}_i = \mathbf{y}_j, \mathbf{i} \in V(X), \mathbf{j} \in V(Y) \\ \alpha_E & : \mathbf{x}_i = \mathbf{y}_j, \mathbf{i} \in E(X), \mathbf{j} \in E(Y) \\ \alpha_{\bar{E}} & : \mathbf{i} \in \bar{E}(X), \mathbf{j} \in \bar{E}(Y) \\ 0 & : \text{otherwise} \end{cases} \quad (6)$$

for all items \mathbf{i} of X and \mathbf{j} of Y . We require that the parameters α_V , α_E , and $\alpha_{\bar{E}}$ are nonnegative such that $\alpha_V + \alpha_E + \alpha_{\bar{E}} > 0$. Thus, exact matching problems only credit exact correspondences between items of X and Y . Standard formulations of the problems listed in Table 4.1 aim at maximizing the cardinality of vertices of the common substructure. Thus, we may set $\alpha_V = 1$ and $\alpha_E = \alpha_{\bar{E}} = 0$. If we want to maximize the cardinality of edges of the common substructure, we may set $\alpha_E = 1$ and $\alpha_V = \alpha_{\bar{E}} = 0$. Other choices of the parameters α_V , α_E , and $\alpha_{\bar{E}}$ reflect the importance of an exact association between corresponding items.

Different types of exact graph matching problems differ from the definition of the search space \mathcal{M} as indicated in Table 4.1. Note that the special cases considered here are GMPs constrained over subsets of total morphisms. Clearly, subsets of total morphisms are not \mathfrak{p} -closed. Hence, in a strict sense, the special cases are not equivalent to clique search. To argue consistently, we regard a special case as an instance of the corresponding generic case, where we make an additional decision based on the optimal solution. For example, we think of the isomorphism problem as a MCISP, where we decide that both graphs under consideration are isomorphic if they have the same cardinality of vertices as a maximum clique in a derived association graph. Using this convention, we can show that the examples in Table 4.1 are equivalent to clique search in an association graph.

Corollary 1. *Consider the GMP (5) with compatibility values of the form (6). Then the problems listed in Table 4.1 are equivalent to the MWCP in an association graph.*

Proof. From Proposition 1 follows that the sets of partial subgraph-, iso-, and homomorphisms are \mathfrak{p} -closed. For problems, which are maximized over total morphisms, we may relax \mathcal{M} to partial morphisms without affecting the optimal solutions of (5), since α_V , α_E , and $\alpha_{\bar{E}}$ are nonnegative with $\alpha_V + \alpha_E + \alpha_{\bar{E}} > 0$. The assertion follows from Theorem 1. \square

4.2 Inexact Graph Matching Problems

In this subsection, we consider inexact graph matching problems. Again let X and Y be graphs with adjacency matrices $\mathbf{X} = (x_{ij})$ and $\mathbf{Y} = (y_{ij})$.

Best Common Subgraph Problems

The *best common subgraph problem* is a GMP (5), where the compatibility values of the matching objective are arbitrary real values. The search space \mathcal{M} is either the set of all partial morphisms or the set of all partial monomorphisms from X to Y .

Corollary 2. *Let X and Y be attributed graphs. Then the best common subgraph problem is equivalent to the MWCP in $Z = X \diamond Y$.*

Proof. Proposition 1 and Theorem 1. □

Probabilistic Graph Matching Problem

In probabilistic graph matching, a probability model is drawn to measure compatibility between items. The aim is to then find a morphism from X to Y that maximizes a global maximum a posteriori probability. To describe the probabilistic graph matching problem in formal terms, we first introduce a distinguished *null color* ϵ_V for vertices not contained in \mathcal{A} . Next, we extend the model Y by including an isolated vertex with null color ϵ_V . The *probabilistic graph matching problem* is defined by

$$\begin{aligned} & \text{maximize} && f(\phi, X, Y) = P(\phi | \mathbf{X}, \mathbf{Y}) \\ & \text{subject to} && \phi \in \mathcal{M}_{XY}, \end{aligned} \tag{7}$$

where the matching objective $P(\phi | \mathbf{X}, \mathbf{Y})$ is the a posteriori probability of ϕ given the *measurements* \mathbf{X} and \mathbf{Y} .² Applying Bayes Theorem, we obtain

$$P(\phi | \mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{X}, \mathbf{Y} | \phi) P(\phi)}{p(\mathbf{X}, \mathbf{Y})}, \tag{8}$$

where $P(\phi)$ is the joint prior for ϕ . The quantities $p(\mathbf{X}, \mathbf{Y} | \phi)$ and $p(\mathbf{X}, \mathbf{Y})$ are the conditional measurement density and the probability density functions, respectively, for the sets of measurements.

Probabilistic graph matching problems are usually solved by a Bayesian inference scheme that does not require explicit calculation of compatibility values in advance. Conceptually, there are compatibility values and a probabilistic graph matching problem turns out to be a GMP (5) over the set of all total morphisms, where we extend the graph Y by an isolated vertex with void attribute ϵ_V .

Corollary 3. *The probabilistic graph matching problem of X and Y is equivalent to the MWCP in $Z = X \diamond Y'$, where Y' extends Y by an isolated vertex with void attribute ϵ_V .*

Proof. Proposition 1 and Theorem 1. □

Graph Edit Distance Problem

The concept of graph edit distance generalizes the Levenshtein edit distance originally defined for strings [13]. The graph edit-distance is defined as the minimum cost over all sequences of basic edit operations that transform X into Y . Following common use, the set of basic edit operations are *substitution*, *insertion*, and *deletion* of items. Different cost functions can be assigned to each edit operation.

The sequences of edit operations that make X and Y isomorphic can be identified with the partial monomorphisms $\phi : V(X) \rightarrow V(Y)$. Each partial monomorphism ϕ induces a bijection $\phi : \mathcal{D}(\phi) \mapsto \mathcal{R}(\phi)$ from the domain $\mathcal{D}(\phi)$ to the range $\mathcal{R}(\phi)$ of ϕ . In terms of ϕ , the edit operations have the following form

²In accordance with the terminology used in [21] we refer to \mathbf{X} and \mathbf{Y} as the sets of measurements.

- *Substitution*: An item \mathbf{i} from $\mathcal{D}(\phi)$ is *substituted* by item \mathbf{i}^ϕ from $\mathcal{R}(\phi)$.
- *Deletion*: Items \mathbf{i} of $\overline{\mathcal{D}}(\phi) = I(X) \setminus \mathcal{D}(\phi)$ are *deleted* from X .
- *Insertion*: Items \mathbf{j} from $\overline{\mathcal{R}}(\phi) = I(Y) \setminus \mathcal{R}(\phi)$ are *inserted* into Y .

If $x_i = y_{i^\phi}$, the substitution \mathbf{i} by \mathbf{i}^ϕ is called *identical substitution*. The cost of a partial monomorphism ϕ is then defined by

$$f(\phi, X, Y) = \sum_{\mathbf{i} \in \overline{\mathcal{D}}(\phi)} C_{del}(\mathbf{i}) + \sum_{\mathbf{j} \in \overline{\mathcal{R}}(\phi)} C_{ins}(\mathbf{j}) + \sum_{\mathbf{i} \in \mathcal{D}(\phi)} C_{sub}(\mathbf{i}, \mathbf{i}^\phi), \quad (9)$$

where $C_{del}(\mathbf{i})$ is the cost of deleting item \mathbf{i} of X , $C_{ins}(\mathbf{i})$ is the cost of inserting an item \mathbf{i} into Y , and $C_{sub}(\mathbf{i}, \mathbf{j})$ is the cost of substituting an item \mathbf{i} from X by an item \mathbf{j} of Y . We assume that all costs are nonnegative.

The *graph edit distance problem* is then of the form

$$\begin{aligned} & \text{minimize} && f(\phi, X, Y) \\ & \text{subject to} && \phi \in \mathcal{M} \end{aligned} \quad (10)$$

where the matching objective f is defined as in (9) and \mathcal{M} is the subset of all partial monomorphisms from X to Y . The constrained global maximum of $-f$ is the *graph edit distance* of X and Y .

To show equivalence between the graph edit distance problem and clique search, we transform problem (10) to our standard form of a GMP as given in (5). First, we expand the set \mathcal{A} of attributes to $\mathcal{A}' = \mathcal{A} \cup \{\mathbf{d}\}$ by including the distinguished symbol \mathbf{d} . We call items with attribute \mathbf{d} *dummy items*. Next, we expand X and Y by adding dummy vertices. Suppose that X and Y are of order $|X| = n$ and $|Y| = m$, respectively. Insert m dummy vertices into X and n dummy vertices into Y . Connected each dummy vertex with all the other (original and dummy) vertices by dummy edges. Let X' and Y' be the resulting expanded graphs. We call X' and Y' the *dummy extensions* of X and Y . Finally, we define an appropriate compatibility function κ . To this end, we first introduce some auxiliary notations. By $V(X, a)$ we denote the subset of all vertices of X that have attribute a . Furthermore, let C'_{sub} be a nonnegative real-valued function on $I(X') \times I(Y')$ such that

$$C'_{sub}(\mathbf{i}, \mathbf{j}) = \begin{cases} C_{sub}(\mathbf{i}, \mathbf{j}) & : \mathbf{i} \in I(X), \mathbf{j} \in I(Y) \\ 0 & : \text{otherwise} \end{cases}.$$

Now we consider the following compatibility values of X' and Y'

$$\kappa_{ij} = \begin{cases} -C_{del}(\mathbf{i}) & : \mathbf{i} \in V(X), \mathbf{j} \in V(Y', \mathbf{d}) \\ -C_{ins}(\mathbf{j}) & : \mathbf{i} \in V(X', \mathbf{d}), \mathbf{j} \in V(Y) \\ -C'_{sub}(\mathbf{i}, \mathbf{j}) & : \text{otherwise} \end{cases} \quad (11)$$

for all items \mathbf{i} of X' and \mathbf{j} of Y' . Then the GMP over the set of all total monomorphisms from X' to Y' is equivalent to the graph edit distance problem (10). Note that we consider negative costs as compatibility values to turn the minimization problem (10) into a maximization problem as in (5).

Corollary 4. *Let X and Y be attributed graphs. Then the graph edit distance problem is equivalent to the MWCP in $Z = X' \diamond Y'$, where X' and Y' are the dummy extensions of X and Y .*

Proof. Proposition 1 and Theorem 1. □

Solving the graph edit distance by clique search in $Z = X' \diamond Y'$ is impractical, because the dummy extensions double the total number of vertices of both original graphs. In a practical implementation, it is sufficient to expand each of both graphs X and Y by adding only one dummy vertex as described above. This leads to a formulation of the graph matching problem constrained over relations $\phi \subseteq V(X) \times V(Y)$ rather than partial morphisms. Though the theory developed in this contribution also holds for relations, we do not consider the more general case for the sake of clarity.

5 Conclusion

We presented a necessary and sufficient condition (C) for graph matching to be equivalent to the maximum weight clique problem in an association graph. The implications of this result are as follows: first, equivalence proofs now reduce to showing that condition (C) holds; second, the condition (C) is applicable to a broad range of common graph matching problems; third, generic continuous solutions can now be applied to all graph matching problems that satisfy (C).

We showed that the graph edit distance problem, probabilistic graph matching, the best common graph matching, and the maximum common homomorphic subgraph problem are equivalent to clique search in a derived association graph.

One limitation with this framework motivates further research: The maximum weight clique problem, where weights are assigned to both vertices and edges has been rarely studied in the literature. Hence, it is conducive to develop special continuous formulations and solutions to the maximum weight clique problem in order to obtain a generic graph matching solver.

Literatur

- [1] A.P. Ambler, H.G. Barrow, C.M. Brown, R.M. Burstall, and R. J. Popplestone. A versatile computer-controlled assembly system. In *International Joint Conference on Artificial Intelligence*, pages 298–307. Stanford University, California, 1973.
- [2] H. Barrow and R. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters*, 4:83–84, 1976.
- [3] M. Bartoli, M. Pelillo, K. Siddiqi, and S.W. Zucker. Attributed tree homomorphism using association graphs. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 2133–2136, 2000.

- [4] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689–694, 1997.
- [5] C.-W.K. Chen and D.Y.Y. Yun. Toward solving maximal overlap set problems. Technical Report TR-LIPSC&Y96a, Laboratory of Intelligent and Parallel Systems, University of Hawaii, 1996.
- [6] C.-W.K. Chen and D.Y.Y. Yun. Unifying graph-matching problem with a practical solution. In *Proceedings of International Conference on Systems, Signals, Control, Computers*, 1998.
- [7] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.
- [8] D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.
- [9] R.L. Goldstone. Similarity, interactive activation, and mapping. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(3):3–28, 1994.
- [10] B.J. Jain. *Structural Neural Learning Machines*. Berlin University of Technology, 2005.
- [11] M. Johnson. Some constraints on embodied analogical understanding. In D.H. Helman, editor, *Analogical Reasoning. Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy*. Kluwer Academic Publishers, 1988.
- [12] V. Kann. On the approximability of NP-complete optimization problems. Master’s thesis, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, 1992.
- [13] V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Doklady*, 10:707–710, 1966.
- [14] G. Levi. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo*, 9:341–352, 1972.
- [15] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11(8):1933–1955, 1999.
- [16] M. Pelillo. Matching free trees, maximal cliques, and monotone game dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1535–1541, 2002.
- [17] M. Pelillo, K. Siddiqi, and S.W. Zucker. Attributed tree matching and maximum weight cliques. In *Proc. ICIAP’99-10th Int. Conf. on Image Analysis and Processing*, pages 1154–1159. IEEE Computer Society Press, 1999.

- [18] M. Pelillo, K. Siddiqi, and S.W. Zucker. Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1120, 1999.
- [19] J.W. Raymond, E.J. Gardiner, and P. Willett. RASCAL: Calculation of graph similarity using maximum common edge subgraphs. *Computer Journal*, 45(6):631–644, 2002.
- [20] K. Schädler and F. Wysotzki. Comparing structures using a Hopfield-style neural network. *Applied Intelligence*, 11:15–30, 1999.
- [21] R.C. Wilson and E.R. Hancock. Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):634–648, 1997.