On the complexity of digraph packings

Richard C. Brewster*† Romeo Rizzi‡§

April 2, 2004

Abstract

Let $\mathcal{G}$ be a fixed collection of digraphs. Given a digraph H, a $\mathcal{G}$-packing of H is a collection of vertex disjoint subgraphs of H, each isomorphic to a member of $\mathcal{G}$. For undirected graphs, Loebl and Poljak have completely characterized the complexity of deciding the existence of a perfect $\mathcal{G}$-packing, in the case that $\mathcal{G}$ consists of two graphs one of which is a single edge on two vertices. We characterize $\mathcal{G}$-packing where $\mathcal{G}$ consists of two digraphs one of which is a single arc on two vertices.

Keywords: Computational complexity, graph algorithms, graph packings.

1 Introduction

We consider finite graphs and digraphs. Let $\mathcal{G}$ be a fixed collection of digraphs. Given a digraph H, a *$\mathcal{G}$-packing* of H is a collection of vertex disjoint subgraphs of H, each isomorphic to a member of $\mathcal{G}$. For graphs, this notion is a natural generalization of matchings, since a matching can be viewed as a collection of vertex disjoint copies of K_2, i.e. a $\{K_2\}$-packing. Given a packing, a vertex of H is *covered* if it belongs to one of the members in the packing. A packing is *perfect* if all vertices are covered. The *$\mathcal{G}$-packing problem* is the problem of deciding for a given input digraph H, whether or not H admits a perfect $\mathcal{G}$-packing.

The $\mathcal{G}$-packing problem has received much attention [2, 3, 5, 6, 7, 9] in the case of undirected graphs. In particular, [3] and [6] recognized the importance of hypomatchable graphs, and [9] continued this work by completely classifying the computational

*The authors gratefully acknowledge the support of the NSERC Canada.

†Computer Science Department, Bishop's University, Lennoxville, Quebec, Canada, J1M 1Z7, rbrewste@ubishops.ca

‡Financial support from project TMR-DONET nr. ERB FMRX-CT98-0202 of the European Community.

§Dipartimento di Informatica e Telecomunicazioni, Università di Trento, via Sommarvie 14, 38050 Povo, Italy, romeo@science.unitn.it

complexity of packing with the family $\mathcal{G} = \{K_2, G\}$, i.e. a single edge and a graph G, where G is connected. For a recent survey see [8].

In the case of digraphs, packing with directed paths and cycles have been examined in [1]. The purpose of this note is to present a complete classification of the computational complexity of the $\{\vec{P}_1, D\}$-packing problem, i.e. a single arc and a digraph D, where D is weakly connected. (We use $\vec{P}_i$ to denote the directed path of length i.)

We make extensive use of the following definitions. A graph G is *perfectly matchable*, or has a *perfect matching*, if it admits a perfect $\{K_2\}$-packing. A graph G is *hypomatchable* if $G - v$ has a perfect matching for every vertex v of G. A graph G is a *propeller* if it can be obtained from a hypomatchable graph B by adding a new pair c, r of vertices, some new edges connecting c with a nonempty subset of vertices of B, and the edge cr. The graph B is the *blade* of the propeller. The vertex c is the *centre*, and the vertex r is the *root*. In [9] it is shown for a fixed, connected graph G, the problem of determining whether a given input graph admits a perfect $\{K_2, G\}$-packing is polynomial time solvable if G is perfectly matchable; G is hypomatchable; or G is a propeller; and the problem is NP-complete otherwise.

The *underlying graph* of a digraph $D = (V, A)$ is the graph on vertex set V where two vertices are adjacent if and only if they are the ends of some arc in A. Given a digraph D, we call D respectively *perfectly matchable*, *hypomatchable*, or *a propeller* if the underlying graph of D is respectively perfectly matchable, hypomatchable, or a propeller. Our main result is the following.

Theorem 1.1 *Let D be a fixed, weakly connected digraph. The $\{\vec{P}_1, D\}$-packing problem is polynomial time solvable if D is perfectly matchable, hypomatchable, or $\vec{P}_2$. The problem is NP-complete in all other cases.*

(The assumption that D is weakly connected is not required for the positive results: the result holds for all perfectly matchable digraphs D; moreover, hypomatchable digraphs must be weakly connected.)

We begin by observing that a digraph packing problem is NP-complete whenever the underlying graph packing problem is NP-complete.

Proposition 1.2 *Let $\mathcal{G}'$ be a collection of digraphs, and let $\mathcal{G}$ be the collection of underlying graphs of $\mathcal{G}'$. That is, $\mathcal{G} = \{G : G$ is the underlying graph of some $D \in \mathcal{G}'\}$. Suppose $\mathcal{G}$-packing is NP-complete. Then $\mathcal{G}'$-packing is NP-complete.*

Proof: Clearly $\mathcal{G}'$-packing is in NP. We reduce $\mathcal{G}$-packing to $\mathcal{G}'$-packing. Let H be an instance of $\mathcal{G}$-packing. Let H' be the symmetric digraph defined by $V(H') = V(H)$

and $(u,v),(v,u) \in E(H')$ if and only if $\{u,v\} \in E(H)$. The digraph H' admits a perfect $\mathcal{G}'$-packing if and only if H admits a perfect $\mathcal{G}$-packing. ▮

Corollary 1.3 *Let D be a weakly connected digraph. The $\{\vec{P}_1, D\}$-packing problem is NP-complete if D is not perfectly matchable, hypomatchable, or a propeller.*

Proof: The result follows directly from the Loebl, Poljak classification [9] for graphs and Proposition 1.2. ▮

In the next sections, to complete the proof of Theorem 1.1, we will examine $\{\vec{P}_1, D\}$-packing in the case that D is perfectly matchable, hypomatchable, or a propeller.

2 Some preliminary cases

Suppose D is a perfectly matchable digraph. Then any $\{\vec{P}_1, D\}$-packing of a digraph, say H, can avoid the use of D. Given any packing $\mathcal{P}$ of H, we can simply replace each copy of $D \in \mathcal{P}$ with a perfect $\{\vec{P}_1\}$-packing of D. The packing thus obtained covers the same set of vertices in H as $\mathcal{P}$. The following proposition is immediate.

Proposition 2.1 *Let D be a perfectly matchable digraph. Then the $\{\vec{P}_1, D\}$-packing problem is polynomially time solvable.*

Consider the case that $D = \vec{P}_2$. In [1] a reduction of $\{\vec{P}_1, \vec{P}_2\}$-packing to bipartite matching is presented. Thus the problem is polynomial time solvable. For any propeller other than $\vec{P}_2$, the $\{\vec{P}_1, D\}$-packing problem is NP-complete. This is shown in Section 4. The remaining case (D is hypomatchable) is presented in Section 3.

3 Hypomatchable digraphs

Let G be a fixed hypomatchable graph. In [2, 5, 6] the $\{K_2, G\}$-packing problem, as well as the following restricted version of the problem, are shown to be polynomial time solvable.

Restricted $\{K_2, G\}$-packing.

Instance: A graph H and a set $\mathcal{C}$ of subgraphs of H each isomorphic to G.

Question: Does H admit a perfect $\{K_2, G\}$-packing where each copy of G in the packing belongs to $\mathcal{C}$?

Proposition 3.1 *Let D be a hypomatchable digraph. The $\{\vec{P}_1, D\}$-packing problem is polynomial time solvable.*

Proof: Let H be an instance of $\{\vec{P}_1, D\}$-packing. We reduce the digraph $\{\vec{P}_1, D\}$-packing problem to the (undirected) restricted $\{K_2, G\}$-packing problem, where G is the underlying graph of D. Let $\mathcal{C}$ be all subgraphs of H isomorphic to D. (Note D is fixed, and thus $|\mathcal{C}|$ is polynomial in the order of H.) The instance of the restricted $\{K_2, G\}$-packing problem is the underlying graph of H, say H', and the collection $\mathcal{C}' = \{C' : C' \text{ is the underlying graph of some } C \in \mathcal{C}\}$. The collection $\mathcal{C}'$ is precisely those copies of G in H' that correspond to copies of D in H. Hence there is a natural correspondence between perfect $\{\vec{P}_1, D\}$-packings of H and restricted $\{K_2, G\}$-packings of H'. ▮

4 Propellers

We now show that the $\{\vec{P}_1, P\}$-packing problem is NP-complete if P is a propeller different from the directed path of length two. To this end, let P be such a propeller. Let r be the root of P, c be the centre, and n be a neighbour of c in the blade of P.

Let E be a boolean formula in conjunctive normal form. Let $X = \{x_1, \ldots, x_n\}$ be the set of variables and $D = \{d_1, \ldots, d_m\}$ be the set of clauses in E. The following restricted version of SAT is NP-complete (see Problem A 9.1. [L01] on page 259 [4]):

(A) at most 3 literals per clause;
(B) every variable occurs in at most 3 clauses.

We may assume every clause has at least two literals, for a clause with one literal uniquely determines the value of that variable in any satisfying truth assignment and allows for a natural reduction where restrictions (A) and (B) are both mantained.

Moreover, if all occurrences of a variable are positive, then we can set that variable to true and drop that variable and all clauses containing it. By inverting a variable, we can also assume that for each variable:

(C) either the variable has one positive occurrence and one negated occurrence;
(D) or the variable has two positive occurrences and one negated occurrence.

For the variable x_i we will label the corresponding literals with $x_i, \overline{x}_i$ in case (C), and with $x_i^1, x_i^2, \overline{x}_i$ in case (D).

Thus, let E be a boolean expression with the restrictions (A), (B), (C) and (D) described above. We construct a digraph $H = H(E)$ from *truth-setting components* and *testing components*. There is one testing component for each clause in E and one truth-setting component for each variable in E. The digraph $H(E)$ will admit a perfect $\{\vec{P}_1, P\}$-packing if and only if E admits a satisfying truth assignment.

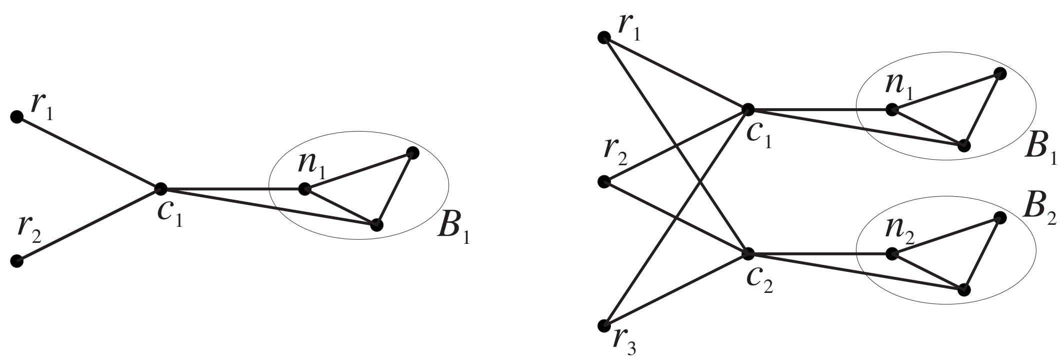

Figure 1: Examples of the testing component

If P has a trivial blade, we will exploit the following assumption: the number of arcs between the root and the centre does not exceed the number of arcs between the blade and the centre. (If necessary, reverse the role of the root and the blade.)

Construct a vertex for each literal appearing in E. These vertices, called *literal vertices*, play a special role in that all components are connected only by means of these vertices. More precisely, a literal vertex will belong to the truth setting component of the involved variable and to the testing component of the clause in which the literal appears.

Let d be a clause in E. We construct a testing component, i.e. a digraph, which we call TC_d. Let l_1, l_2 and possibly l_3 be the k literals appearing in d. Construct $k-1$ copies of P, say $D_1, \ldots, D_{k-1}$. We label the blade of D_i with B_i, the centre with c_i, and the root with r_i. The root is the literal vertex corresponding to l_i; hence l_i and r_i label the same vertex. In addition, add arcs between the pairs (l_i, c_j), $i \neq j$, so that any l_i together with any $c_j \cup B_j$ is a copy of P with root l_i. Examples of the testing components are shown in Figure 1.

The main property and basic role of the testing component is revealed by the following lemma.

Lemma 4.1 *Let S be a* non-empty *subset of $\{l_1, \ldots, l_k\}$. Then there exists a $\{\vec{P}_1, P\}$-packing covering all vertices of TC_d except those in S. Moreover, TC_d admits no perfect $\{\vec{P}_1, P\}$-packing.*

The truth setting components are now defined. Let x_i be a variable occurring in E. If x_i occurs twice, then the truth-setting component TS_i associated with x_i consists of a directed path of length 2. The two ends of the path are the two literal vertices and hence are labelled with x_i and $\overline{x}_i$; the centre of the path is labelled v_i. Observe that no $\{\vec{P}_1, P\}$-packing of the truth-setting component can cover both x_i and $\overline{x}_i$ since P differs from $\vec{P}_2$. If x_i occurs three times in E, then the truth-setting

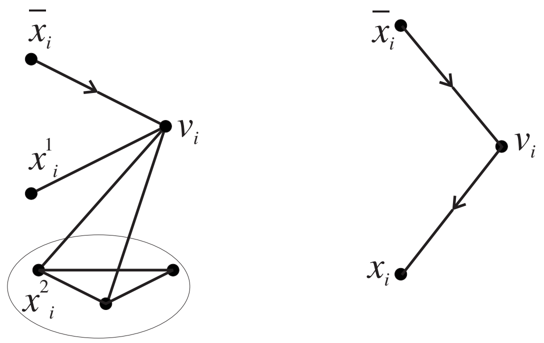

Figure 2: Examples of the truth-setting components

component is constructed as follows: Begin with a copy P' of P where x_i^1 is the root and x_i^2 is the specified neighbour, n, of the centre. Label the centre with v_i. If there is an arc of P' with tail x_i^1 and head v_i, then add a single arc from v_i to $\overline{x}_i$; otherwise, add a single arc from $\overline{x}_i$ to v_i. An example of each kind of truth-setting component appears in Figure 2.

Lemma 4.2 *Any packing of TS_i which covers all of $TS_i - \{\overline{x}_i, x_i^1, x_i^2\}$ cannot cover both $\overline{x}_i$ and a vertex labelled x_i^j.*

Proof: Assume $\overline{x}_i$ is covered by a packing of TS_i which covers all of $TS_i - \{\overline{x}_i, x_i^1, x_i^2\}$. We show that $\overline{x}_i$ is not covered by a propeller. By our choice of the single arc between v_i and $\overline{x}_i$, no subgraph of TS_i isomorphic to P can have root $\overline{x}_i$. Clearly, such a subgraph cannot have centre $\overline{x}_i$ either. For $\overline{x}_i$ to belong to its blade, P has a trivial blade, and P has a single arc between the centre and the blade. Since we assumed that the number of arcs between the root and the centre does not exceed the number of arcs between the blade and the centre, and since P differs from $\vec{P}_2$, it finally follows that $\overline{x}_i$ cannot be covered by a copy of P inside TS_i. Therefore, $\overline{x}_i$ must be covered by a copy of $\vec{P}_1$ which also covers v_i. Hence, the packing does not cover x_i^1. Moreover, the packing does not cover x_i^2 either, since the blade of P is hypomatchable, and the packing covers all of $TS_i - \{\overline{x}_i, x_i^1, x_i^2\}$. ∎

This completes the construction of $H(E)$. That is, $H(E)$ consists of the literal vertices upon which we have constructed testing components and truth-setting components.

Lemma 4.3 *Let $H(E)$ be the digraph constructed as above and suppose $\mathcal{P}$ is a perfect $\{\vec{P}_1, P\}$-packing of $H(E)$. Assume a node c_j of a testing component is covered by a propeller in $\mathcal{P}$. Then c_j is covered by the centre of the propeller.*

Proof: If c_j is covered by the root of a propeller, then the hypomatchable graph B_j must be covered by copies of $\vec{P}_1$, a contradiction. If c_j belongs to the blade of a propeller, say B', then B' must cover an odd number of vertices of B_j. The removal of c_j from B' disconnects the blade B' creating a component with an odd number of vertices. This cannot happen in a hypomatchable graph. Consequently, c_j must be covered by the centre of B'. ∎

The result in the following lemma is related to the notion of *coherence* introduced in [7].

Lemma 4.4 *Let $\mathcal{P}$ be a perfect packing of $H(E)$. Then there exists a perfect packing $\mathcal{P}'$ of H such that each member of $\mathcal{P}'$ is a subgraph of some truth-setting component or a subgraph of some testing component.*

Proof: Clearly every $\vec{P}_1$ in $\mathcal{P}$ is a subgraph of some truth-setting component or a subgraph of some testing component. In light of Lemma 4.3, the result also holds if P has a blade consisting of a single vertex.

Thus assume P has a nontrivial blade. Let P' be a copy of P which covers vertices c_j and l_i in some testing component TC_d, but is not a subgraph of TC_d. By Lemma 4.3, we know that the centre of P' covers c_j. Since the blade of P', say B', is connected either all of B_j is covered by B' or none of B_j is covered by any of B'. In the former case P' is a subgraph of TC_d. Thus the latter case must occur. Since B_j cannot be covered solely by copies of $\vec{P}_1$, the root of P' must cover a vertex in B_j. The remainder of B_j is covered by copies of $\vec{P}_1$.

We will now show that B' covers a blade in the truth-setting component to which l_i belongs. The vertex l_i is covered by a vertex in the blade B'. Since B' is a nontrivial hypomatchable graph, it does not contain any vertex of degree one. Hence l_i has degree at least two in B'. We conclude that l_i is x_q^2 in truth-setting component TS_q. (Note that l_i may also be adjacent to another centre in TC_d; however, by Lemma 4.3 this centre cannot be covered by any vertex in B'.) Let B_q denote the blade in TS_q to which x_q^2 belongs.

As in the proof of Lemma 4.3 we can see that all of B_q must be covered by B'. Otherwise, the removal of the v_q from B' would leave a component with an odd number of vertices in B'. This cannot happen in a hypomatchable graph.

We now remove P' and the copies of $\vec{P}_1$ which cover B_j from $\mathcal{P}$. We add a propeller covering $x_q^2 = l_i$ with its root, c_j with its centre, and B_j with its blade. We cover the remainder of B_k with copies of $\vec{P}_1$. The result follows. ∎

Theorem 4.5 *The $\{\vec{P}_1, P\}$-packing problem is NP-complete.*

Proof: Let E be an instance of SAT with the restrictions above. The digraph $H = H(E)$ is the instance of the $\{\vec{P}_1, P\}$-packing problem.

Suppose there is a satisfying truth assignment for E. For each variable x_j, if x_j is false, then we cover $\overline{x}_j, v_j$ with $\vec{P}_1$. If x_j is true, then we cover $TS_j - \overline{x}_j$ with $\vec{P}_1$ (respectively P) if x_j appears twice (respectively three times) in E. For each testing component TC_d, at least one of the vertices labelled with l_i must be covered at this point. The remainder of the testing component can be covered by copies of $\vec{P}_1$ and P.

On the other hand, suppose H has a perfect $\{\vec{P}_1, P\}$-packing. By Lemma 4.4 we can assume each member in the packing is a subgraph of a truth-setting component or a testing component. The first direction of the proof provides a template for determining the truth assignment for the variables. Each testing component must have at least one vertex l_i that is not covered by a subgraph of the testing component. Thus l_i must be covered by a subgraph of some truth-setting component. Specifically, the vertex l_i is some $\overline{x}_j$ (in which case we assign false to x_j) or it is one of $\{x_j, x_j^1, x_j^2\}$ (in which case we assign true to x_j). By our above remarks and Lemma 4.2, it is not the case that both $\overline{x}_j$ and one of $\{x_j, x_j^1, x_j^2\}$ are covered by subgraphs of the truth-setting component. Consequently, no variable is assigned both true and false. We conclude that our assignment is a satisfying truth assignment for E. ∎

References

[1] R. C. Brewster, P. Hell, S. H. Pantel, R. Rizzi, and A. Yeo, Packing paths in digraphs, submitted.

[2] G. Cornuéjols, D. Hartvigsen, and W. Pulleyblank, Packing subgraphs in a graph, *Operations Research Letters,* **1**(1982), 139 – 143.

[3] G. Cornuéjols and D. Hartvigsen, An extension of matching theory, *Journal of Combinatorial Theory, Series B,* **40**(1986), 285 – 296.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W. H. Freeman, San Fransisco, 1979.

[5] P. Hell and D. G. Kirkpatrick, On generalized matching problems, *Information Processing Letters,* **12**(1981), 33 – 35.

[6] P. Hell and D. G. Kirkpatrick, Packings by cliques and by finite families of graphs, *Discrete Mathematics,* **49**(1984), 45 – 59.

[7] D. G. Kirkpatrick and P. Hell, On the complexity of general graph factor problems, *SIAM Journal on Computing,* **12**(1983), 601 – 609.

[8] P. Hell. Graph Packings. *Electronic Notes in Discrete Mathematics, Proceedings of the 6th International Conference on Graph Theory Marseille (France), August 28 - September 1, 2000*. Editor Irena Rusu **5**(2000)

[9] M. Loebl and S. Poljak, Efficient subgraph packing, *Journal of Combinatorial Theory, Series B,* **59**(1993), 106 – 121.