The P_k partition problem and related problems in bipartite graphs

Jérôme Monnot ⋆ and *Sophie Toulouse* ⋆⋆

No Institute Given

Abstract. In this paper, we continue the investigation proposed in [15] about the approximability of $\mathbf{P}_k$ partition problems, but focusing here on their complexity. More precisely, we prove that the problem consisting of deciding if a graph of nk vertices has n vertex disjoint simple paths $\{P_1, \cdots, P_n\}$ such that each path P_i has k vertices is **NP**-complete, even in bipartite graphs of maximum degree 3. Note that this result also holds when each path P_i is chordless in $G[V(P_i)]$. Then, we prove that the optimization version of these problems, denoted by Max$\mathbf{P}_3$Packing and MaxInduced$\mathbf{P}_3$Packing, are not in **PTAS** in bipartite graphs of maximum degree 3. Finally, we propose a 3/2-approximation for Min3-PathPartition in general graphs within $O(nm + n^2 \log n)$ time and a 1/3 (resp., 1/2)-approximation for MaxW$\mathbf{P}_3$Packing in general (resp., bipartite) graphs of maximum degree 3 within $O(\alpha(n, 3n/2)n)$ (resp., $O(n^2 \log n)$) time, where α is the inverse Ackerman's function and $n = |V|$, $m = |E|$.

Keywords: P_k-partition; Induced P_k-partition; maximum (weighted) P_k-packing; maximum (weighted) induced P_k-packing; minimum k-path partition; bipartite graphs; **NP**-completeness; **APX**-hardness; approximation algorithms.

1 Introduction

The $\mathbf{P}_k$ partitioning problem ($\mathbf{P}_k$Partition in short) consists, given a simple graph $G = (V, E)$ on $k \times n$ vertices, of deciding if there exists a partition of V into $(V_1, \cdots, V_n)$ such that for $1 \leq i \leq n$, $|V_i| = k$ and the subgraph $G[V_i]$ induced by V_i contains a Hamiltonian path. In other words, we want to know if there exists n vertex disjoint simple paths of length k in G. The analogous problem where the subgraph $G[V_i]$ induced by V_i is isomorphic to $\mathbf{P}_k$ (the chordless path on k vertices) will be denoted by induced $\mathbf{P}_k$Partition. These two problems are **NP**-complete for any $k \geq 3$, and polynomial otherwise, [8, 13]. In fact, they both are a particular case of a more general problem called *partition into isomorphic subgraphs*, [8]. In [13], Kirkpatrick and Hell give a necessary and

⋆ CNRS LAMSADE - UMR 7024, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France. monnot@lamsade.dauphine.fr

⋆⋆ LIPN - UMR CNRS 7030, Institut Galilée, Université Paris 13, 99 av. Jean-Baptiste Clément, 93430 Villetaneuse, France. sophie.toulouse@lipn.univ-paris13.fr

sufficient condition for the **NP**-completeness of the partition into isomorphic subgraphs problem in general graphs.

$\mathbf{P}_k$Partition has been widely studied in the literature, mainly because its **NP**-completeness also implies the **NP**-hardness of two famous optimization problems, namely: the minimum k-path partition problem (denoted by Min$k$-PathPartition) and the maximum $\mathbf{P}_k$ packing problem (Max$\mathbf{P}_k$Packing in short). Min$k$-PathPartition consists of partitioning the vertex set of a graph $G = (V, E)$ into the smallest number of paths such that each path has *at most* k vertices (for instance, Min2-PathPartition is equivalent to the edge cover problem); the optimal value is usually denoted by $\rho_{k-1}(G)$, and by $\rho(G)$ when no constraint occurs on the length of the paths (in particular, we have $\rho(G) = 1$ *iff* G has an Hamiltonian path). Min$k$-PathPartition has been extensively studied in the literature, [19, 18, 22], and has applications in broadcasting problems, see for example [22]. Max$\mathbf{P}_k$Packing (resp., MaxInduced$\mathbf{P}_k$Packing), consists, given a simple graph $G = (V, E)$, of finding a maximum number of vertex-disjoint (resp., induced) $\mathbf{P}_k$. In their weighted versions (denoted MaxW$\mathbf{P}_k$Packing and MaxWInduced$\mathbf{P}_k$Packing, respectively), the input graph $G = (V, E)$ is given together with a weight function $w : E \to \mathbb{N}$ on its edges; the goal is to find a collection $\mathcal{P} = \{P_1, \ldots, P_q\}$ of vertex-disjoint (resp., induced $\mathbf{P}_k$) maximizing $w(\mathcal{P}) = \sum_{i=1}^{q} \sum_{e \in P_i} w(e)$. Some approximation results for MaxW$\mathbf{P}_k$Packing when the graph is complete on $k \times n$ vertices are given in [9, 10, 15]. In this case, each solution contains exactly n vertex disjoints paths of length $k-1$ (note that, in this particular case, the minimization version may also be considered). This problem is related to the vehicle routing problem, [22, 3].

Here, we study the complexity of $\mathbf{P}_k$Partition and induced $\mathbf{P}_k$Partition in the case of bipartite graphs. We first show that $\mathbf{P}_k$Partition and induced $\mathbf{P}_k$Partition are **NP**-complete for any $k \geq 3$ in bipartite graphs of maximum degree 3. Moreover, for $k = 3$, this remains true even if the graph is planar. On the opposite, $\mathbf{P}_k$Partition, induced $\mathbf{P}_k$Partition, Min$k$-PathPartition and MaxW$\mathbf{P}_k$Packing trivially become polynomial-time computable in graphs of maximum degree 2 and in forests. Then, we prove that, in bipartite graphs of maximum degree 3, Max$\mathbf{P}_k$Packing and MaxInduced$\mathbf{P}_k$Packing are not in **PTAS**. More precisely, we prove that there is a constant $\varepsilon_k > 0$ such that it is **NP**-hard to decide whether a maximum (induced) $\mathbf{P}_k$-packing of a bipartite graph of maximum degree 3 on kn vertices is of size n or of size upper bounded by $(1-\varepsilon_k)n$. Finally, we propose a 3/2-approximation for Min3-PathPartition in general graphs and a 1/3 (resp., 1/2)-approximation for MaxW$\mathbf{P}_3$Packing in general (resp., bipartite) graphs of maximum degree 3.

This paper is organized as follows: in the next section, we briefly present previous related works about the hardness of solving bounded-size-path packing problems. Then, the third part is dedicated to complexity results concerning the problems $\mathbf{P}_k$Partition, induced $\mathbf{P}_k$Partition, MaxInduced$\mathbf{P}_k$Packing and Max$\mathbf{P}_k$Packing in bipartite graphs. Finally, some approximation results concerning MaxW$\mathbf{P}_3$Packing and Min3-PathPartition are proposed in a fourth section. A full version of this paper is appeared as the technical report, [16].

The notations are the usual ones according to graph theory. Moreover, we exclusively work on undirected simple graphs. In this paper, we often identify a path P of length $k-1$ with $\mathbf{P}_k$, even if P contains a chord. However, when we deal with INDUCED $\mathbf{P}_k$PARTITION, the paths considered will be chordless. We denote by $opt(I)$ and $apx(I)$ the value of an optimal and of an approximate solution, respectively. We say that an algorithm $\mathcal{A}$ is an ε-approximation with $\varepsilon \geq 1$ for a minimization problem (resp., with $\varepsilon \leq 1$ for a maximization problem) if $apx(I) \leq \varepsilon \times opt(I)$ (resp., $apx(I) \geq \varepsilon \times opt(I)$) for any instance I (for more details, see for instance [2]).

2 Previous related work

The minimum k-path partition problem is obviously **NP**-complete in general graphs [8], and remains intractable in comparability graphs, [19], in cographs, [18], and in bipartite chordal graphs, [19] (when k is part of the input). Note that most of the proofs of **NP**-completeness actually establish the **NP**-completeness of $\mathbf{P}_k$PARTITION. Nevertheless, the problem turns out to be polynomial-time solvable in trees, [22], in cographs when k is fixed, [18] or in bipartite permutation graphs, [19]. Note that one can also find in the literature several results about partitioning the graph into disjoints paths of length at least 2, [20, 11].

Concerning the approximability of related problems, Hassin and Rubinstein, [9] proposed a generic algorithm to approximate MAXW$\mathbf{P}_4$PACKING in complete graphs on $4n$ vertices that guarantees an approximation ratio of $3/4$ for general distance function. More recently in [15], it has been proven that this algorithm is also a $9/10$-approximation for the $1,2$-instances. For the minimization version, it provides respectively a $3/2$- and a $7/6$-approximation for the metric and the $1,2$- instances in complete graphs on $4n$ vertices (in this case, we seek a maximal $\mathbf{P}_4$-Packing of minimum weight). In [10], the authors proposed a $(35/67-\varepsilon)$-approximation for MAX$\mathbf{P}_3$PARTITION in complete graphs on $3n$ vertices using a randomized algorithm. To our knowledge, there is no specific approximation results for MAXW$\mathbf{P}_3$PACKING in general graphs. However, using approximation results for the maximum weighted 3-packing problem (mainly based on local search techniques), [1], we can obtain a $(\frac{1}{2}-\varepsilon)$-approximation for MAXW$\mathbf{P}_3$PACKING. Finally, there is, to our knowledge, no approximation result for MINk-PATHPARTITION. Nevertheless, when the problem consists of maximizing the number of edges used by the paths, then we can find some approximation results, in [21] for the general case, in [5] for dense graphs.

3 Complexity results

Theorem 1. $\mathbf{P}_k$PARTITION *and* INDUCED $\mathbf{P}_k$PARTITION *are* ***NP****-complete in bipartite graphs of maximum degree 3, for any $k \geq 3$. As a consequence, the problems* MAX$\mathbf{P}_k$PACKING *and* MINk-PATHPARTITION *are* ***NP****-hard in bipartite graphs with maximum degree 3, for any $k \geq 3$.*

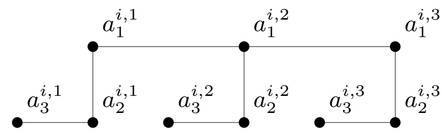

Fig. 1. The gadget $H(c_i)$ when c_i is a 3-tuple.

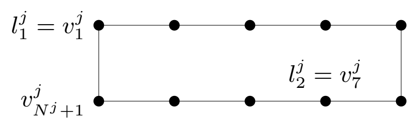

Fig. 2. The gadget $H(e_j)$ for $k = 3$ and $d^j = 2$.

Proof. (sketch). The proof is based on a reduction from the k-dimensional matching problem, denoted by kDM, which is known to be **NP**-complete, [8]. Since the paths of length $k-1$ that are used in this reduction are chordless, the result holds for both $\mathbf{P}_k$PARTITION and INDUCED $\mathbf{P}_k$PARTITION. An instance of kDM consists of a subset $\mathcal{C} = \{c_1, \ldots, c_m\} \subseteq X_1 \times \ldots \times X_k$ of k-tuples, where $X_1, \ldots, X_k$ are k pairwise disjoint sets of size n. A matching is a subset $M \subseteq \mathcal{C}$ such that no element in M agrees in any coordinate, and the purpose of kDM is to answer the question: does there exist a perfect matching M on $\mathcal{C}$, that is, a matching of size n?

We first do the proof for odd values of k. Given an instance $I = (\mathcal{C}, X_1 \times \ldots \times X_k)$ of kDM, we build an instance $G = (V, E)$ of $\mathbf{P}_k$PARTITION, where G is a bipartite graph of maximum degree 3, as follows:

• To each k-tuple $c_i \in \mathcal{C}$, we associate a gadget $H(c_i)$ that consists of a collection $\left\{P^{i,1}, \ldots, P^{i,k}\right\}$ of k vertex-disjoint $\mathbf{P}_k$ with $P^{i,q} = \left\{a_1^{i,q}, \ldots, a_k^{i,q}\right\}$ for $q = 1, \ldots, k$. We add to $H(c_i)$ the edges $[a_1^{i,q}, a_1^{i,q+1}]$ for $q = 1$ to $k-1$, in order to form a $(k+1)$-th $\mathbf{P}_k$ $\left\{a_1^{i,1}, \ldots, a_1^{i,k}\right\}$ (see Figure 1 for an illustration when $k = 3$).

• For each element $e_j \in X_1 \cup \ldots \cup X_k$, let d^j denote the number of k-tuples $c_i \in \mathcal{C}$ that contain e_j; the gadget $H(e_j)$ is defined as a cycle $\left\{v_1^j, \ldots, v_{N^j+1}^j, v_1^j\right\}$ on $N^j + 1$ vertices, where $N^j = k(2d^j - 1)$. Moreover, for $p = 1$ to d^j, we denote by l_p^j the vertex of index $2k(p-1)+1$ (see Figure 2 for an illustration of $H(e_j)$ when $k = 3$ and $d^j = 2$).

• Finally, for any couple (e_j, c_i) such that e_j is the value of c_i on the q-th coordinate, the two gadgets $H(c_i)$ and $H(e_j)$ are connected using an edge $[a_2^{i,q}, l_{p_i}^j]$. The vertices $l_{p_i}^j$ that will be linked to a given gadget $H(c_i)$ must be chosen in such a way that each vertex l_p^j from any gadget $H(e_j)$ will be connected to exactly one gadget $H(c_i)$ (this is possible since each $H(e_j)$ contains exactly d^j vertices l_p^j).

This construction leads to a graph on $3k^2m + (1-k)kn$ vertices: consider, on the one hand, that each gadget $H(c_i)$ is a graph on k^2 vertices and, on the

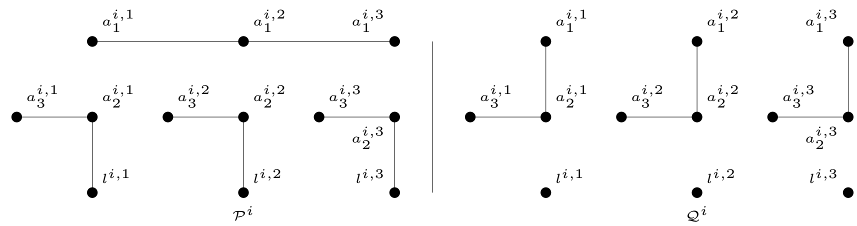

Fig. 3. Two possible vertex partitions of a $H(c_i)$ gadget into 2-length paths.

other hand, that $\sum_{j=1}^{kn} d^j = km$ (wlog., we assume that each element e_j appears at least once in $\mathcal{C}$). Finally, G is obviously bipartite of maximum degree 3. We claim that there exists a perfect matching $M \subseteq \mathcal{C}$ *iff* there exists a partition $\mathcal{P}^*$ of G into $\mathcal{P}_k$. The following property can be easily proved:

Property 1. In any partition of G into $\mathcal{P}_k$, and for any $i = 1, \ldots, m$, one uses either $\mathcal{P}^i$ or $\mathcal{Q}^i$, where $\mathcal{P}^i$ and $\mathcal{Q}^i$ are the collections of paths defined as:

$$\forall i = 1, \ldots, m, \forall q = 1, \ldots, k, \begin{cases} P^{i,q} = \left\{a_k^{i,q}, \ldots, a_2^{i,q}, l_{i,q}\right\} \\ Q^{i,q} = \left\{a_k^{i,q}, \ldots, a_2^{i,q}, a_1^{i,q}\right\} \end{cases}$$

(where $l_{i,q}$ denotes the vertex from some $H(e_j)$ linked to $a_2^{i,q}$).

$$\forall i = 1, \ldots, m, \qquad \begin{cases} \mathcal{P}^i = \cup_{q=1}^{k} P^{i,q} \cup \left\{a_1^{i,1}, a_1^{i,2}, \ldots, a_1^{i,k}\right\} \\ \mathcal{Q}^i = \cup_{q=1}^{k} Q^{i,q} \end{cases}$$

Let M be a perfect matching on $\mathcal{C}$; we build a packing $\mathcal{P}$ applying the following rule: if a given element c_i belongs to M, then we use $\mathcal{P}^i$ to cover $H(c_i)$, and we use $\mathcal{Q}^i$ otherwise; Figure 3 illustrates this construction for 3DM. Since M is a perfect matching, exactly one vertex l_p per gadget $H(e_j)$ is already covered by some $P^{i,q}$. Thus, on a given cycle $H(e_j)$, the $N^j = k(2d^j - 1)$ vertices that remain uncovered can easily be covered using a sequence of $(2d^j - 1)$ vertex disjoints $\mathcal{P}_k$. Conversely, let $\mathcal{P}^* = \{P_1, \ldots, P_r\}$ be a partition of G into $\mathbf{P}_k$; since each gadget $H(e_j)$ has $N^j = k(2d^j - 1) + 1$ vertices, at least one edge e of some P_ℓ in $\mathcal{P}^*$ links $H(e_j)$ to a given $H(c_i)$, using a l_p vertex; we deduce from Property 1 that P_ℓ is some $P^{i,q}$ path and thus, that l_p is the only vertex from $H(e_j)$ that intersects P_ℓ. Consider now any two vertices l_p and $l_{p'}$, $p < p'$, from $H(e_j)$; since $l_p = v_{2k(p-1)+1}$ and $l_{p'} = v_{2k(p'-1)+1}$, there are $2k(p' - p) - 1$ vertices between l_p and $l_{p'}$, which might not be covered by any collection of $\mathcal{P}_k$. Hence, exactly one vertex from each $H(e_j)$ is covered by some $P^{i,q}$. Concerning $H(c_i)$, we already know that its vertices may be covered by either $\mathcal{P}^i$, or $\mathcal{Q}^i$; hence, by setting $M = \left\{c_i \mid \mathcal{P}^i \subseteq \mathcal{P}^*\right\}$, we define a perfect matching, and the proof is complete.

The proof is quite identical for even values of k. The only difference lies on the $H(e_j)$ gadgets, that consists of a cycle $\left\{v_1^j, \ldots, v_{N^j}^j, v_1^j\right\}$ on N^j vertices, plus an additional edge $[v_{N^j}^j, v_{N^j+1}^j]$.

If we decrease the maximum degree of the graph down to 2, we can easily prove that $\mathbf{P}_k$Partition, induced $\mathbf{P}_k$Partition, Max$\mathbf{P}_k$Packing and Min$k$-PathPartition are polynomial-time computable. The same fact holds for MaxW$\mathbf{P}_k$Packing, although it is a little bit complicated. Moreover, this result holds in forests.

Proposition 1. MaxW$\mathbf{P}_k$Packing *is polynomial in graphs with maximum degree 2 and in forests, for any* $k \geq 3$.

Proof. (sketch.) We reduce the problem of computing an optimal solution of MaxW$\mathbf{P}_k$Packing in graphs with maximum degree 2 (or in a forest) to the problem of computing a maximum weight independent set (MaxWIS in short) in an interval (or chordal) graph, which is known to be polynomial, [7]. The reduction that is made is the usual one when dealing with set packing problems: from an instance of MaxW$\mathbf{P}_k$Packing, we construct a graph $G' = (V', E')$ where V' is isomorphic to the set of $\mathbf{P}_k$ of the initial graph and where E' describes the intersection relation between the $\mathbf{P}_k$; the weight associated to a vertex from V' is naturally set to the weight of the $\mathbf{P}_k$ this vertex represents.

On the other hand, if we restrict our attention to planar bipartite graphs of maximum degree 3, $\mathbf{P}_3$Partition and induced $\mathbf{P}_3$Partition remain intractable.

Theorem 2. $\mathbf{P}_3$Partition *and* induced $\mathbf{P}_3$Partition *are* ***NP****-complete in planar bipartite graphs with maximum degree 3. As a consequence,* Max$\mathbf{P}_3$Packing *and* Min3-PathPartition *are* ***NP****-hard in planar bipartite graphs with maximum degree 3.*

Proof. (sketch). The construction made in Theorem 1 transforms an instance of the planar 3-dimensional matching problem (Planar 3DM-3 in short), which is still **NP**-complete, [6], into a planar graph (just note that the choice of the vertex l_p^j from $H(e_j)$ that will be linked to $H(c_i)$ is no longer free, but depends on the characteristic graph of the input instance of Planar 3DM-3).

Lemma 1. *For any* $k \geq 3$*, there is a constant* $\varepsilon_k > 0$*, such that* $\forall G = (V, E)$ *instance of* Max$\mathbf{P}_k$Packing *(resp.,* MaxInduced$\mathbf{P}_k$Packing*) where* G *is a bipartite graph of maximum degree 3, it is* ***NP****-complete to decide whether* $opt(G) = \frac{|V|}{k}$ *or* $opt(G) \leq (1 - \varepsilon_k)\frac{|V|}{k}$*, where* $opt(G)$ *is the value of a maximum (resp., maximum induced)* $\mathbf{P}_k$*-Packing on* G*.*

Proof. (sketch). The argument is based on an **APX**-hardness result concerning the optimization version of kDM (denoted by Max$k$DM): for any $k \geq 3$, there exists a constant $\varepsilon'_k > 0$, such that $\forall I = (\mathcal{C}, X_1 \times \ldots \times X_k)$ instance of Max$k$DM with $n = |X_q|\ \forall q$, it is **NP**-complete to decide whether $opt(I) = n$ or $opt(I) \leq (1 - \varepsilon'_k)n$, where $opt(I)$ is the value of a maximum matching on $\mathcal{C}$. Furthermore, this result also holds if we restrict our attention to instances of Max$k$DM with bounded degree, namely, to instances verifying $d^j \leq f(k)\ \forall j$, where $f(k)$ is a constant (we refer to [17] for $k = 3$, to [12] for other values of k). Let I be an instance of Max$k$DM such that $\forall e_j \in X_1 \cup \ldots \cup X_k$, $d^j \leq f(k)$. Consider the graph $G = (V, E)$ produced in Theorem 1. We recall that G is bipartite, of maximum degree 3, on $|V| = 3k^2m + (1-k)n$ vertices (where $m = |\mathcal{C}|$). Furthermore, all paths of length $k-1$ in G are chordless. Let $\mathcal{P}^*$ be an optimal solution of Max$\mathbf{P}_k$Packing with value $opt(G)$. The argument lies on that we can assume wlog. the following two facts:

(i) For any k-uple c_i, $\mathcal{P}^*$ contains either the packing $\mathcal{P}^i$, or the packing $\mathcal{Q}^i$ of the gadget $H(c_i)$.
(ii) For any element e_j, $\mathcal{P}^*$ contains exactly $2d^j - 1$ paths from the gadget $H(e_j)$.

Under these assumptions, if m_0 denotes the number of elements c_i such that $\mathcal{P}^*$ contains $\mathcal{P}^i$, we observe that $opt(I) = m_0$ and thus, we have: $opt(G) = (3km - kn) + opt(I)$. Hence, deciding whether $opt(I) = n$ or $opt(I) \leq (1 - \varepsilon'_k)n$ and deciding whether $opt(G) = (3km - kn) + n$ or $opt(G) \leq (3km - kn) + (1 - \varepsilon'_k)n$ are equivalent. By setting $\varepsilon_k = \frac{n}{3km - kn + n}\varepsilon'_k$, we have $(3km - kn) + (1 - \varepsilon'_k)n = (1 - \varepsilon_k)(3km - kn + n)$. Finally, since $d^j \leq f(k)$ where $f(k)$ is a constant, we deduce that $km \leq kf(k)n$ and then, $\varepsilon_k \geq \frac{1}{3kf(k)+1-k}\varepsilon'_k$, which completes the proof. The **APX**-hardness immediately follows.

Some interesting questions concern the complexity of $\mathbf{P}_k$Partition (or induced $\mathbf{P}_k$Partition) for $k \geq 4$ in planar bipartite graphs with maximum degree 3 and the **APX**-hardness of Max$\mathbf{P}_k$Packing and MaxInduced$\mathbf{P}_k$Packing (or MaxInduced$\mathbf{P}_k$Packing) for $k \geq 3$ in planar bipartite graphs with maximum degree 3.

4 Approximation results

We present some approximation results for the problems MaxW$\mathbf{P}_3$Packing and Min3-PathPartition, that are mainly based on matching and spanning tree heuristics.

4.1 MaxWP$_3$Packing in graphs of maximum degree 3

For this problem, the best approximate algorithm known so far provides a ratio of $(\frac{1}{2} - \varepsilon)$, within high (but polynomial) time complexity. This algorithm is deduced from the one proposed in [1] to approximate the weighted k-set packing problem for sets of size 3. Furthermore, a simple greedy $1/k$-approximation of MaxW$\mathbf{P}_k$Packing consists of iteratively picking a path of length $k-1$ that is of maximum weight. For $k = 3$ and in graphs of maximum degree 3, the time complexity of this algorithm is between $O(n \log n)$ and $O(n^2)$ (depending on the encoding structure). Actually, in such graphs, one may reach a 1/3-approximate solution, even in time $O(\alpha(n, m)n)$, where α is the inverse Ackerman's function and $m \leq 3n/2$.

Theorem 3. MaxW$\mathbf{P}_3$Packing *is 1/3 approximable within* $O(\alpha(n, 3n/2)n)$ *time complexity in graphs of maximum degree 3; this ratio is tight for the algorithm we analyze.*

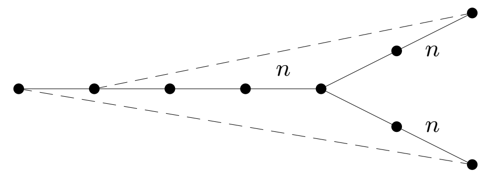

Fig. 4. The tightness.

Proof. We assume that the graph is connected (otherwise, we apply the same proof on each connected component containing at least 3 vertices). The argument lies on the following observation: for any spanning tree of maximum degree 3 containing at least 3 vertices, one can build a cover of its edge set into 3 packings of $\mathbf{P}_3$ within linear time (a formal proof is given in appendix). Hence, given a weighted connected graph $G = (V, E)$ of maximum degree 3, we compute a maximum-weight spanning tree $T = (V, E_T)$ on G. Because G is of maximum degree 3, this can be done in $O(\alpha(n, 3n/2)n)$ time, [4]. We then compute $(\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3)$ a $\mathbf{P}_3$-packing cover of T and finally, pick the best $\mathbf{P}_3$-packing among $\mathcal{P}^1$, $\mathcal{P}^2$ and $\mathcal{P}^3$. The value of this packing is at least $1/3$ times the weight of T, which is at least the weight of an optimal P_3-Packing on G, since any P_3-Packing can be extended into a spanning tree. The tightness of this algorithm is illustrated in Figure 4: the edges of E_T are drawn in rigid lines, whereas the edges of $E \backslash E_T$ are drawn in dotted lines; finally, all the edges with no mention of their weight are of weight 1. Observe that an optimal $\mathbf{P}_3$-packing on T is of weight $n + 3$, whereas $opt(I) = 3n + 3$.

For the unweighted case, we easily see that an optimal $\mathbf{P}_3$-packing uses at most $2|V|/3$ edges. Moreover, computing a spanning tree can be done in linear time, and we can prove that the 3 packing outputted by the solution cover at least $|V|$ vertices. Thus, using Theorem 3, we deduce:

Corollary 1. MAX$\mathbf{P}_3$PACKING *is 1/2 approximable within linear time complexity in graphs of maximum degree 3.*

4.2 MaxWP$_3$Packing in bipartite graphs of maximum degree 3

If we restrict our attention to bipartite graphs, we slightly improve the ratio of $\frac{1}{2} - \varepsilon$ ([1]) up to $\frac{1}{2}$. We then show that, in the unweighted case, this result holds without any constraint on the graph maximum degree.

¿From $I = (G, w)$ where G is a bipartite graph $G = (L \cup R, E)$ of maximum degree 3, we build two weighted graphs (G_L, d_L) and (G_R, d_R), where $G_L = (L, E_L)$ and $G_R = (R, E_R)$. Two vertices $x \neq y$ from L are linked in G_L *iff* there exists in G a path of length 2 $P_{x,y}$ from x to y, rigorously: $[x, y] \in E_L$ *iff* $\exists z \in R$ s.t. $[x, z], [z, y] \in E$. The distance $d_L(x, y)$ is defined as $d_L(x, y) = \max\{w(x, z) + w(z, y) | [x, z], [z, y] \in E\}$. (G_R, d_R) is defined by considering R instead of L. If G is of maximum degree 3, then the following fact holds:

Lemma 2. *From any matching M on G_L (resp., on G_R), one can deduce a $\mathbf{P}_3$ packing $\mathcal{P}_M$ of weight $w(\mathcal{P}_M) = d_L(M)$ (resp., $w(\mathcal{P}_M) = d_R(M)$), when G is of degree at most 3.)*

Proof. We only prove the result for G_L. Let M be a matching on G_L. For any edge $e = [x, y] \in M$, there exists in G a chain $P_e = \{x, z_e, y\}$ with $w(P_e) = d_L(e)$. Let us show that $\mathcal{P}_M = \{P_e | e \in M\}$ is a packing. Assume the reverse: then, there exists two edges $e_1 = [x_1, y_1]$ and $e_2 = [x_2, y_2]$ in M such that $P_{e_1} \cap P_{e_2} \neq \emptyset$. Since $\{e_1, e_2\}$ is a matching, the four vertices x_1, x_2, y_1 and y_2 are pairwise distinct and then necessarily $z_{e_1} = z_{e_2}$. Hence, z_{e_1} is linked to 4 vertices in G, which contradicts the fact that the maximum degree in G does not exceed 3.

`Weighted` P_3`-Packing`

1 Build the weighted graphs (G_L, d_L) and (G_R, d_R);
2 Compute a maximum weight matching M_L^* (resp., M_R^*) on (G_L, d_L) (resp., on (G_R, d_R));
3 Deduce from M_L^* (resp., M_R^*) a $\mathbf{P}_3$ packing $\mathcal{P}_L$ (resp., $\mathcal{P}_R$) according to Lemma 2;
4 Output the best packing $\mathcal{P}$ among $\mathcal{P}_L$ and $\mathcal{P}_R$.

The time complexity of this algorithm is mainly the time complexity of computing a maximum weight matching in graphs of maximum degree 9, that is $O(|V|^2 \log |V|)$, [14].

Theorem 4. *`Weighted` $\mathbf{P}_3$`-Packing` provides a $1/2$-approximation for the problem* MaxW$\mathbf{P}_3$Packing *in bipartite graphs with maximum degree 3 and this ratio is tight.*

Proof. Let $\mathcal{P}^*$ be an optimum P_3-packing on $I = (G, w)$, we denote by $\mathcal{P}_L^*$ (resp., $\mathcal{P}_R^*$) the paths of $\mathcal{P}^*$ of which the two endpoints belong to L (resp., R); thus, $opt(I) = w(\mathcal{P}_L^*) + w(\mathcal{P}_L^*)$. For any path $P = P_{x,y} \in \mathcal{P}_L^*$, $[x, y]$ is an edge from E_L, of weight $d_L(x, y) \geq w(P_{x,y})$. Hence, $M_L = \{[x, y] | P_{x,y} \in \mathcal{P}_L^*\}$ is a matching on G_L that satisfies:

$$d(M_L) \geq w(\mathcal{P}_L^*) \tag{1}$$

Moreover, since M_L^* is a maximum weight matching on G_L, we have $d_L(M_L) \leq d_L(M_L^*)$. Thus, using inequality (1) and Lemma 2 (and by applying the same arguments on G_R), we deduce:

$$w(\mathcal{P}_L) \geq w(\mathcal{P}_L^*),\ w(\mathcal{P}_R) \geq w(\mathcal{P}_R^*) \tag{2}$$

Finally, the solution outputted by the algorithm satisfies $w(\mathcal{P}) \geq 1/2(w(\mathcal{P}_L) + w(\mathcal{P}_R))$; thus, we directly deduce from inequalities (2) the expected result. The instance $I = (G, w)$ that provides the tightness is depicted in Figure 5. It consists of a graph on $12n$ vertices on which one can easily observe that $w(\mathcal{P}_L) = w(\mathcal{P}_R) = 2n(n + 2)$ and $w(\mathcal{P}^*) = 2n(2n + 2)$.

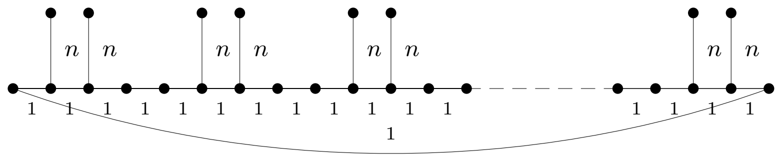

Fig. 5. The tightness.

Concerning the unweighted case, we may obtain the same performance ratio without the restriction on the maximum degree of the graph. The main differences with the previous algorithm lies on the construction of the two graphs G_L, G_R: starting from G, we duplicate each vertex $r_i \in R$ by adding a new vertex r'_i with the same neighborhood as r_i (this operation, often called *multiplication of vertices* in the literature, is used in the characterization of perfect graphs). Finally, we add the edge $[r_i, r'_i]$. If R_L denotes the vertex set $\{r_i, r'_i | r_i \in R\}$, then the following property holds:

Property 2. From any matching M on G_L, one can deduce a matching M' on G_L that saturates R_L, and such that $|M'| \geq |M|$.

Let M be a matching on G_L. If none of the two vertices r_i and r'_i for some i are saturated by M, then set $M' = M \cup \{[r_i, r'_i]\}$. If exactly one of them is saturated by a given edge e from M, then set $M' = (M \setminus \{e\}) \cup \{[r_i, r'_i]\}$. In any case, M' is still a matching of size at least $|M|$. Thus, the expected result is obtained by applying this process to each vertex of R_L.

Theorem 5. *There is a 1/2-approximation for* MAX$\mathbf{P}_3$PACKING *in bipartite graphs and this ratio is tight. The complexity time of this algorithm is* $O(m\sqrt{n})$.

4.3 Min3-PathPartition in general graphs

To our knowledge, the approximability of MINk-PATHPARTITION (or MINPATH-PARTITION) has not been studied so far. Here, we propose a 3/2-approximation for MIN3-PATHPARTITION. Although this problem can be viewed as an instance of 3-set cover (interpret the set of all paths of length 0,1, or 2 in G as sets on V), MIN3-PATHPARTITION and the minimum 3-set cover problem are different. For instance, consider a star $K_{1,2n}$; the optimum value of the corresponding 3-set cover instance is n, whereas the optimum value of the 3-path partition is $2n-1$. Note that, concerning MINPATHPARTITION (that is, the approximation of $\rho(G)$), we can trivially see that it is not $(2-\varepsilon)$-approximable, from the fact that deciding whether $\rho(G) = 1$ or $\rho(G) \geq 2$ is **NP**-complete. Actually, we can more generally establish that $\rho(G)$ is not in **APX**: otherwise, we could obtain a **PTAS** for the traveling salesman problem with weight 1 and 2 when $opt(I) = n$, which is not possible, unless **P**=**NP**.

`Computing` $\rho_2(G)$

1 Compute a maximum matching M_1^* on G;
2 Build a bipartite graph $G_2 = (L, R; E_2)$ where $L = \{l_e | e \in M_1^*\}$, $R = \{r_v | v \in V \setminus V(M_1^*)\}$, and $[l_e, r_v] \in E_2$ *iff* the corresponding isolated vertex $v \notin V(M_1^*)$ is adjacent in G to the edge $e \in M_1^*$;
3 Compute a maximum matching M_2^* on G_2;
4 Output $\mathcal{P}'$ the 3-paths partition deduced from M_1^*, M_2^*, and $V \setminus V(M_1^* \cup M_2^*)$. Precisely, if $M_1' \subseteq M_1^*$ is the set of edges adjacent to M_2^*, then the paths of length 2 are given by $M_1' \cup M_2^*$, the paths of length 1 are given by $M_1^* \setminus M_1'$, and the paths of length 0 (that is, the isolated vertices) are given by $V \setminus V(M_1^* \cup M_2^*)$;

The time complexity of this algorithm is $O(nm + n^2 \log n)$, [14].

Theorem 6. MIN3-PATHPARTITION *is* $3/2$*-approximable in general graphs; this ratio is tight for the algorithm we analyze.*

Proof. (sketch.) Let $G = (V, E)$ be an instance of MIN3-PATHPARTITION. Let $\mathcal{P}^* = (\mathcal{P}_2^*, \mathcal{P}_1^*, \mathcal{P}_0^*)$ be an optimal solution on G, where $\mathcal{P}_i^*$ denotes for $i = 0, 1, 2$ the set of paths of length i. By construction of the approximate solution, we have:

$$apx(I) = |V| - |M_1^*| - |M_2^*| \tag{3}$$

We consider a subgraph $G_2' = (L, R'; E_2')$ of G_2 where R' and E_2' are defined as: $R' = \{r_v \in R | v \notin \mathcal{P}_0^*\}$ and E_2' contains the edges $[l_e, r_v] \in E_2$ such that v is adjacent to e via an edge that belongs to the optimal solution. By construction of G_2', from the optimality of M_1^*, and because $\mathcal{P}^*$ is a 3-path packing, we deduce that $d_{G_2'}(r) \geq 1$ for any $r \in R'$, and that $d_{G_2'}(l) \leq 2$ for any $l \in L$, where $d_{G_2'}(v)$ is the degree of vertex v in graph G_2'. Hence, G_2' contains a matching that is of size at least one-half $|R'|$ and thus:

$$|M_2^*| \geq 1/2|R'| = 1/2\left(|V| - 2|M_1^*| - |\mathcal{P}_0^*|\right) \tag{4}$$

Using inequalities (3) and (4), and considering that $|V| = 3|\mathcal{P}_2^*| + 2|\mathcal{P}_1^*| + 1|\mathcal{P}_0^*|$, we deduce:

$$apx(I) \leq 1/2\left(|V| + |\mathcal{P}_0^*|\right) \text{ and } opt(I) \geq 1/3\left(|V| + |\mathcal{P}_0^*|\right)$$

References

1. E. Arkin, R. Hassin. On local search for weighted packing problems. *Mathematics of Operations Research*, 23: 640-648, 1998.
2. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela and M. Protasi. Complexity and Approximation (Combinatorial Optimization Problems and Their Approximability Properties). *Springer, Berlin*, 1999.

3. C. Bazgan, R. Hassin, and J. Monnot. Approximation algorithms for some routing problems. *Discrete Applied Mathematics*, 146: 3-26, 2005.
4. B. Chazelle. A minimum spanning tree algorithm with Inverse-Ackermann type complexity. *J. ACM*, 47: 1028-1047, 2000.
5. B. Csaba, M. Karpinski, P. Krysta. Approximability of dense and sparse instances of minimum 2-connectivity, TSP and path problems. *SODA*, 74-83, 2002.
6. M. Dyer, A. Frieze. Planar 3DM is NP-complete. *J. Algorithms*, 7:174-184, 1986.
7. A. Frank. Some Polynomial Time Algorithms for Certain Graphs and Hypergraphs. *Proceedings of the 5th British Combinatorial Conference, Congressus Numerantium XV, Utilitas Mathematicae, Winnipeg*, 211-226, 1976).
8. M. R. Garey, D. S. Johnson. Computers and intractability. A guide to the theory of NP-completeness. *CA, Freeman*, 1979.
9. R. Hassin, S. Rubinstein. An Approximation Algorithm for Maximum Packing of 3-Edge Paths. *Inf. Process. Lett.*, 63: 63-67, 1997.
10. R. Hassin, S. Rubinstein. An Approximation Algorithm for Maximum Triangle Packing. *ESA* , LNCS 3221: 403-413, 2004.
11. A. Kaneko. A necessary and sufficient condition for the existence of a path factor every component of which is a path of length at least two. *Journal of Combinatorial Theory, Series B*, 88: 195-218, 2003.
12. M. Karpinski. Personnal communication. 2006.
13. D. G. Kirkpatrick, P. Hell. On the Completeness of a Generalized Matching Problem. *Proc. STOC'78*, 240-245, 1978.
14. L. Lovasz, M. D. Plummer. Matching Theory. *North-Holland,*, Amsterdam, 1986.
15. J. Monnot, S. Toulouse. Approximation results for the weighted P_4 partition problem. *The symposia on Fundamentals of Computation Theory*, F.C.T.'2005, LNCS 3623, 377-385, 2005.
16. J. Monnot, S. Toulouse. The P_k partition problem and related problems in bipartite graphs. *Technical Report* (available at http://www.lamsade.dauphine.fr/ monnot/publications(journal).htm), 2006.
17. E. Petrank. The Hardness of Approximation: Gap Location. *Computational Complexity*, 4, 133-157, 1994.
18. G. Steiner. On the k-Path partition problem in cographs. *Cong. Numer.*, 147:89-96, 2000.
19. G. Steiner. On the k-path partition of graphs. *Theor. Comput. Sci.*, 290:2147-2155, 2003.
20. H. Wang. Path factors of bipartite graphs. *Journal of Graph Theory*, 18: 161-167, 1994.
21. S. Vishwanathan. An Approximation Algorithm for the Asymmetric Travelling Salesman Problem with Distances One and Two. *Information Processing Letter*, 44(6): 297-302, 1992.
22. J-H Yan, G. J. Chang, S. M. Hedetniemi, S.T. Hedetniemi. k-Path partitions in trees. *Discrete Applied Mathematics*, 78:227-233, 1997.