A Parameterized Perspective on Packing Paths of Length Two

Henning Fernau[1] & Daniel Raible[1]

Universität Trier, FB IV—Abteilung Informatik, 54286 Trier, Germany,
{fernau,raible}@informatik.uni-trier.de

Abstract. We study (vertex-disjoint) packings of paths of length two (i.e., of P_2's) in graphs under a parameterized perspective. Starting from a maximal P_2-packing $\mathcal{P}$ of size j we use extremal combinatorial arguments for determining how many vertices of $\mathcal{P}$ appear in some P_2-packing of size $(j+1)$ (if such a packing exists). We prove that one can 'reuse' $2.5j$ vertices. We also show that this bound is asymptotically sharp. Based on a WIN-WIN approach, we build an algorithm which decides, given a graph, if a P_2-packing of size at least k exists in time $\mathcal{O}^*(2.448^{3k})$.

1 Introduction and Definitions

Mathematical Motivation. We consider a natural generalization of the well-known matching problem in graphs. A maximum matching is a maximum cardinality set of vertex disjoint edges, i.e., a packing with paths of length one. We are going to study packings by paths of length two (abbreviated as P_2). More formally, we consider the following problem, called P_2-PACKING:

Given: A graph $G = (V, E)$, and the parameter k.
We ask: Is there a set of k vertex-disjoint P_2's in G?

P. Hell and D. Kirkpatrick [20, 17] proved $\mathcal{NP}$-completeness for this problem. In fact, they showed that general MAXIMUM H-PACKING is $\mathcal{NP}$-complete. Here, H is a graph with at least three vertices in some connected component. Notice that P_2-PACKING attracts attention as it is $\mathcal{NP}$-hard, whereas the classical matching problem, which is P_1-PACKING, is solvable in polynomial time.

Applications.

Test Cover. There is a strong link to the TEST COVER (TC) problem [2] with applications ranging from fault testing and diagnosis, pattern recognition to biological identification. The input to TC is a hypergraph $H = (G, E)$ and one wishes to identify a subset $E' \subseteq E$ (the *test cover*) such that, for any distinct $i, j \in V$, there is an $e' \in E'$ with $|e' \cap \{i, j\}| = 1$. Tests are modeled by hyperedges e; $x \in e$ means that the individual (vertex) x passes the test. TC models identification problems: Given a set of individuals and a set of binary attributes, we search for a minimum subset of attributes that identifies each

individual distinctly. For the special yet important case TCP2, for all $e \in E$, we have $|e| \leq 2$. This means that at most two individuals can pass a certain test. For TCP2, K. M. J. Bontridder *et al.* [2] could show the following two assertions. (1) If H has a test cover of size τ, then there is a P_2-packing of size $n - \tau - 1$ that leaves at least one vertex isolated. (2) If H has a maximal P_2-packing of size π that leaves at least one vertex isolated, then there is a test cover of size $n - \pi - 1$. This also establishes a close relation between TEST COVER and TOTAL EDGE COVER. So we can employ our algorithms to solve the TCP2 case of TEST COVER by using an initial catalytic branch that determines the one vertex that should be isolated.

Vehicle Routing. General packing of paths of length d (called P_d-packing) can be motivated by a vehicle routing problem. Suppose there is a number of customers such that to everyone the same kind of item should be delivered. Delivering is done by the same kind of vehicle. Assume in every such vehicle exactly d items fit in. Now we want to assign the customers to the vehicles in a way that delivery times are minimized. We model this by a graph G: The customers are vertices and two vertices are joined by an edge if their distance is less than some $\epsilon > 0$. In practice ϵ is some reasonable constant which permits to travel from one customer to the other in short time. Now assume we can perfectly pack a set of P_d's in G (perfectly means covering every vertex). Then every P_d corresponds to some vehicle. The vehicle will start at an endpoint and serve every customer along the path. This way we minimize the maximum time a vehicle needs to serve all its assigned customers (not regarding travel time to the first customer site). If d is small (like $d = 2$ in this paper), think of the items being whole containers of goods, but a truck could only deliver up to three containers a time. This approach has been reported in [1, 25], where actually a partitioning (not a packing) of the graph into paths was asked for to model the quest for serving all customers. However, as those problems typically arise on a dynamic day-to-day basis (where the underlying graph also varies day by day), we might ask for such a packing that models the optimal use of all k available trucks on a given day, leaving the remaining vertices (customers) to be delivered the other day, with possible new deliver requests coming up (and hence enlarging the graph again).

Our Framework: Parameterized Complexity. A *parameterized problem* P is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is a fixed alphabet and $\mathbb{N}$ is the set of all non-negative integers. Therefore, each instance of the parameterized problem P is a pair (I, k), where the second component k is called the *parameter*. The language $L(P)$ is the set of all YES-instances of P. We say that the parameterized problem P is *fixed-parameter tractable* [8] if there is an algorithm that decides whether an input (I, k) is a member of $L(P)$ in time $f(k)|I|^c$, where c is a fixed constant and $f(k)$ is a function independent of the overall input length $|I|$. We will also write $\mathcal{O}^*(f(k))$ for this run-time bound. Equivalently, one can define the class of fixed-parameter tractable problems as follows: strive to find a polynomial-time transformation that, given an instance (I, k), produces another instance (I', k')

of the same problem, where $|I'|$ and k' are bounded by some function $g(k)$; in this case, (I', k') is also called a *(problem) kernel*.

Discussion of Related Work. R. Hassin and S. Rubinstein [16] found a randomized $\frac{35}{67}$-approximation for finding a maximum P_2-packing. K. M. J. Bontridder *et al.* [2] studied deterministic approximation algorithms, considering a series of heuristics H_ℓ. H_ℓ starts from a maximal P_2-packing $\mathcal{P}$ and tries to improve it by replacing ℓ P_2's by $\ell + 1$ P_2's. The corresponding approximation ratios ρ_ℓ are as follows: $\rho_0 = \frac{1}{3}$, $\rho_1 = \frac{1}{2}$, $\rho_2 = \frac{5}{9}$, $\rho_3 = \frac{7}{11}$ and $\rho_\ell = \frac{2}{3}$ for $\ell \geq 4$.
As any P_2-PACKING instance can be transformed into a 3-SET PACKING instance one can use the algorithm of Y. Liu *et al.* [23] which needs $\mathcal{O}^*(4.61^{3k})$ steps, or the very recent algorithm of J. Wang and Q. Feng [30] running in time $\mathcal{O}(3.52^{3k})$. This is the culmination point of a sequence of papers subsequently improving on the running time of this problem. Alternatively, we can use randomized parameterized algorithms; I. Koutis [22] has developed a randomized parameterized algorithm for this problem that runs in time $\mathcal{O}^*(2^{3k})$. The first paper to individually study P_2-PACKING under a parameterized view was E. Prieto and C. Sloper [29]. The authors were able to prove a $15k$-kernel. Via a clever midpoint search on the kernel they could achieve a deterministic run time of $\mathcal{O}^*(3.403^{3k})$. Another special case of 3-SET PACKING studied from a parameterized perspective is 3-DIMENSIONAL MATCHING, see [23] for a deterministic algorithm of run time $\mathcal{O}^*(2.77^{3k})$. Recently, J. Wang *et al.* [31] found a kernel of size $7k$ for P_2-PACKING, resulting in a deterministic $\mathcal{O}^*(2.61^{3k})$-algorithm for this problem.

Our Contributions. We mention that the presented results where partly already published in form of a conference paper (H. Fernau, D. Raible [13]) and as a technical report (J. Chen et *al.* [4]). Actually, [13] and [31] are spin-offs of [4]. The main achievements of this paper are:
(1) We present an iterative-augmentation-type algorithm which solves this problem in time $\mathcal{O}^*(2.448^{3k})$. (2) We exhibit an extremal combinatorial argument to show that, given a maximal P_2-packing of size j and provided that a larger packing exists, we can reuse $2.5j$ vertices of the known packing. This improves a similar result for general 3-SET PACKING [23] where only $2j$ elements are known to be reusable. (3) We present a graph family such that the before mentioned reusability result is asymptotically sharp. (4) Another novelty is that in this algorithm, the inductive augmentation step is interleaved with kernelization. This pays off not only heuristically but also asymptotically by a specific form of combinatorial analysis. Thereby we can completely skip the time consuming color-coding which was needed in Liu *et al.* [23] for 3-SET PACKING. (5) We show that WIN-WIN games can be played with two different brute-force algorithms to finally achieve the claimed running time. We believe that especially the idea of saving colors by extremal combinatorial arguments could be applied in other situations, as well. (6) We derive a new linear kernel result for the related problem of TOTAL EDGE COVER and can also give lower bounds on the kernel size.

Additionally, we must admit that in the previous versions of this paper ([13] and [4]) there is a mistake. Due to complexity reasons the search for the endpoint pairs cannot be undertaken. Thus, only a run time of $\mathcal{O}^*(17.44^k)$ can be claimed instead of $\mathcal{O}^*(15.285^k)$. Nevertheless, in this paper we achieve a run time for the problem which beats the one which was originally claimed.

Some Notations and Definitions. We only consider undirected graphs $G = (V, E)$. For a subgraph H of G, denote by $N(H)$ the set of vertices that are not in H but adjacent to at least one vertex in H, i.e., $N(H) = (\bigcup_{v \in H} N(\{v\})) \setminus H$. The subgraph H is *adjacent* to a vertex v if $v \in N(H)$. A P_2 in G is a path which consists of three vertices and two edges. For any path p of this kind we consider the vertices as numbered such that $p = p_1p_2p_3$ (where the roles of p_1 and p_3 might be interchanged). At some points we consider p as a tuple (p_1, p_2, p_3) if necessary. For a path p, $V(p)$ ($E(p)$, resp.) denotes the set of vertices (edges, resp.) of p. Likewise, for a set of paths $\mathcal{P}$, $V(\mathcal{P}) := \bigcup_{p \in \mathcal{P}} V(p)$ ($E(\mathcal{P}) := \bigcup_{p \in \mathcal{P}} E(p)$, resp.). A vertex v is called $\mathcal{P}$-*midpoint* if there is a $p = p_1p_2p_3 \in \mathcal{P}$ with $p_2 = v$. Let $\mathcal{M}_\mathcal{P} := \{p_2 \mid \exists p \in \mathcal{P} : p = p_1p_2p_3\}$ contain the $\mathcal{P}$-midpoints. Vertices from $V(\mathcal{P})$ that are not $\mathcal{P}$-midpoints are called $\mathcal{P}$-*endpoints*.

Organization of the Paper. In Sec. 2, we discuss the notion of parametric duality in the context of the P_2-PACKING problem, deriving lower and upper bounds on the kernel size of TOTAL EDGE COVER. Sec. 3 presents the combinatorial results we derived for P_2-packings, in particular concerning the reusability of vertices and, more specifically, of midpoints in the attempt to construct larger packings from smaller ones. In Sec. 4, we present the iterative augmentation algorithm, including its analysis. Sec. 5 is mostly devoted to displaying examples of graph families that show that new rules and insights are necessary to further improve our algorithm along the lines of thought.

2 Issues From Parameterized Complexity

In parameterized complexity, the *(parametric) dual* of a vertex-selection problem, parameterized by a bound k on the solution size (the *natural parameter* of the problem), is defined by reversing the parameter, i.e., by considering $k_d = |V| - k$ instead of k as the parameter of the problem. This notion should not be confused with the classical notion of duality known from linear programming. However, in analogy with that area, the problem parameterized by k is then called the *primal* problem. For example, the classical VERTEX COVER problem is such a vertex-selection problem. The natural parameter k is upper-bounding the size of an acceptable solution (vertex cover). As it is well-known, a graph $G = (V, E)$ has a vertex cover of size $|V| - k$ if and only if G has an independent set of size k. Hence, VERTEX COVER and INDEPENDENT SET (using the natural parameterizations that bound the solution sets) are (parametric) duals to each other. While VERTEX COVER is in $\mathcal{FPT}$, INDEPENDENT SET (on general

graphs) is likely not to be (both problems considered with the natural parameterizations). Similarly, while DOMINATING SET is fixed-parameter intractable on general graphs, its parametric dual, called NONBLOCKER, is fixed-parameter tractable [7]. In fact, many problems which are parameterized tractable become intractable when the parameter is "turned around" (see [10, 19, 28]). However, knowing that both primal and parametric dual are in $\mathcal{FPT}$ (and hence kernelizable) allows to state lower-bound results on kernel sizes, see [3], which makes such problem pairs very interesting for parameterized complexity.

An *edge cover* is a set of edges $EC \subseteq E$ that cover all vertices of a given graph $G = (V, E)$. A *matching* is a set of edges $M \subseteq E$ of a given graph $G = (V, E)$, where each two edges e, e' from M have no common endpoint. According to the well-known theorem of T. Gallai [14], the size of a maximum matching of a graph $G = (V, E)$ plus the size of a minimum edge cover of G equals $|V|$. In a somewhat generalized fashion as discussed above, this means that EDGE COVER and MATCHING can be viewed as parametric duals. By matching techniques, the problem of finding an edge cover of size at most k is hence solvable in polynomial time, see [24]. (It is known (see [27]) that even MINIMUM WEIGHTED EDGE COVER can be optimally solved in polynomial (cubic) time.)

H. Fernau and D. F. Manlove [12] discovered a similar relation of P_2-PACKING to TOTAL EDGE COVER, which we now describe. An edge cover is called *total* if every component in $G[EC]$ has at least two edges. This type of constraint for covering problems is motivated by modeling clustering properties within cover sets, see [12]. However, the following Gallai-type identity [12] proves that finding total edge covers of size at most k is $\mathcal{NP}$-hard: The sum of the number of P_2's in a maximum P_2-packing and the size of a minimum total edge cover equals $n = |V|$. The proof of the mentioned Gallai-type identity used only the fact that the covers and packings were minimal and maximal, respectively, with respect to (edge) set inclusion, although the statement of the identity refers to minimum and maximum solutions, respectively. Hence, observing that the covers and packings in question could be minimal and maximal, we can infer:

Theorem 1. *Let $G = (V, E)$ be a graph. G contains a P_2-packing of size at least k if and only if there is a total edge cover of size at most $|V| - k$ in G.* □

H. Fernau and D. F. Manlove [12] also showed that TOTAL EDGE COVER is fixed-parameter tractable (or: lies in $\mathcal{FPT}$, for short). Hence, both the P_2-PACKING problem (being in the focus of this paper), asking for a set of k vertex-disjoint P_2's in the given graph $G = (V, E)$, and the question to find $(|V| - k)$ vertex-disjoint P_2's in the given graph $G = (V, E)$, with parameter k, respectively, are in $\mathcal{FPT}$. Equivalently, both the question of finding a total edge cover of size k_d in the given graph $G = (V, E)$ (i.e., TOTAL EDGE COVER) and the question of finding a total edge cover of size $(|V| - k_d)$ in G, with parameter k_d, are in $\mathcal{FPT}$. This is quite interesting since there are few natural, unrestricted problems where both the primal and the dual variant are known to lie in $\mathcal{FPT}$.

We mention here that the (more general results) of H. Fernau and D. Manlove [12] on kernel sizes of variants of edge-cover problems can be improved for the para-

metric dual (in the sense of Theorem 1) TOTAL EDGE COVER, parameterized by k_d upper-bounding the edge cover size:

Theorem 2. TOTAL EDGE COVER *admits a kernel with at most* $1.5k_d$ *vertices.*

Proof. Since we aim at a total edge cover, the largest number of vertices that can be covered by k_d edges is $1.5k_d$ (namely, if the edge cover is a P_2-packing). Hence, if the graph contains more than $1.5k_d$ vertices, we can reject. This leaves us with a kernel with at most $1.5k_d$ vertices. □

Based on the work [29] of E. Prieto and C. Sloper, the authors of [4, 31] exhibited the following result:

Theorem 3 ([31]). P_2-PACKING *admits a kernel with at most* $7k$ *vertices.*

This also allows us to state lower bounds for the kernel sizes, based on works of J. Chen *et al.* [3]:

Corollary 1. *Trivially,* P_2-PACKING *does not admit a kernel with less than* $3k$ *vertices.* TOTAL EDGE COVER *does not admit a kernel with less than* $\alpha_a k_d$ *vertices for any* $\alpha_d < (7/6)$*, unless* $\mathcal{P} = \mathcal{NP}$*.*

Proof. A P_2-packing of size k is only possible in a graph with at least $3k$ vertices. Due to Theorem 3 and [3, Theorem 3.1], there does not exist a kernel of size $\alpha_d k_d$ for TOTAL EDGE COVER under the assumption that $\mathcal{P} \neq \mathcal{NP}$ if $(7-1)(\alpha_d-1) < 1$. □

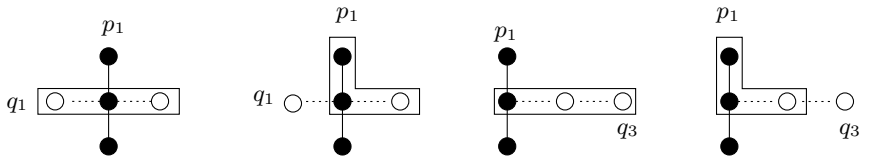
3 Combinatorial Properties of P_2-Packings

This section is devoted to proving the following combinatorial result by extremal combinatorial arguments. Notice that $\mathfrak{Q}_{(2)}$ denotes a specific family of P_2-packings of size $(j+1)$. The exact definition of $\mathfrak{Q}_{(2)}$ will be given later in this section, but its understanding is not necessary to appreciate the algorithmic consequences of this result at this stage: Having obtained (somehow) a maximal P_2-packing $\mathcal{P}$ of size j, it is sufficient to find $(0.5j+3)$ more vertices from $V \setminus V(\mathcal{P})$ to built a P_2-packing $\mathcal{Q}$ of size $(j+1)$, if possible.

Theorem 4. *Let* $\mathcal{P}$ *be a maximal* P_2*-packing of size* j*. If there is a* P_2*-packing of size* $(j+1)$*, then there is also a packing* $\mathcal{Q} \in \mathfrak{Q}_{(2)}$ *with* $|V(\mathcal{P}) \cap V(\mathcal{Q})| \geq 2.5j$*.*

The combinatorial properties of $\mathcal{Q}$ will be used in the next section by the inductive step of our algorithm for P_2-PACKING. So, in our induction hypothesis we assume that we already know a (with respect to inclusion) maximal P_2-packing $\mathcal{P}$. Among all maximal P_2-packings of size $(j+1)$, we will consider those packings that recycle as many paths from $\mathcal{P}$ as possible. More formally, we consider those $\mathcal{Q}$ that maximize

$$|\{p \in \mathcal{P} \mid \exists q \in \mathcal{Q} : E(p) = E(q)\}|. \tag{1}$$



(a) q is foldable on p. (b) (q_1, p_1)-folding. (c) q is shiftable on p. (d) (q_3, p_1)-shifting.

Fig. 1. The black vertices and solid edges indicate the P_2-packing $\mathcal{P}$. The polygons contain the P_2's of the packing $\mathcal{Q}$. Dotted line are not contained in $E(\mathcal{P})$.

We call the set of these packings $\mathfrak{Q}_{(1)}$.[1] From Liu et *al.* [23], we know:

Lemma 1 ([23]). *$|V(p) \cap V(\mathcal{Q})| \geq 2$ for any $p \in \mathcal{P}$ and $\mathcal{Q} \in \mathfrak{Q}_{(1)}$.*

Proof. If there is $p \in \mathcal{P}$ with $|V(p) \cap V(\mathcal{Q})| = 1$, then replace the path $q \in \mathcal{Q}$ that intersects with p by p. In the case where $|V(p) \cap V(\mathcal{Q})| = 0$, simply replace an arbitrary $q \in \mathcal{Q} \setminus \mathcal{P}$, that must exist by pigeon-hole, by p. In both cases, we obtain a packing $\mathcal{Q}'$ of the same size as $\mathcal{Q}$, but recycling more paths, contradicting $\mathcal{Q} \in \mathfrak{Q}_{(1)}$. □

A slightly sharper version is the next assertion:

Corollary 2. *If $\mathcal{Q} \in \mathfrak{Q}_{(1)}$, then for any $p \in \mathcal{P}$ with $p \notin \mathcal{Q}$, there are $q_1, q_2 \in \mathcal{Q}$, $q_1 \neq q_2$, with $|V(p) \cap V(q_i)| \geq 1$ $(i = 1, 2)$.*

Proof. Suppose it exists $p \in \mathcal{P}$ and only one $q \in \mathcal{Q}$ with $|V(p) \cap V(q)| \geq 2$ (Lemma 1). Then $\mathcal{Q} \setminus \{q\} \cup \{p\}$ improves on property (1), contradicting $\mathcal{Q} \in \mathfrak{Q}_{(1)}$. □

We sharpen this combinatorial bound by considering from the set $\mathfrak{Q}_{(1)}$ only those P_2-packings $\mathcal{Q}'$ which maximize the following second property:

$$|E(\mathcal{P}) \cap E(\mathcal{Q}')| \tag{2}$$

The set of the remaining P_2-packings will be called $\mathfrak{Q}_{(2)}$. So, in $\mathfrak{Q}_{(2)}$ are those packings from $\mathfrak{Q}_{(1)}$ which share the maximum number of edges from $\mathcal{P}$.

In contrast to the general situation with 3-SET PACKING, paths are more concrete objects that can be shifted or folded along the given graph. These geometric ideas will be used to finally prove our claimed combinatorial theorem. We formalize these intuive notions of folding and shifting of paths in the following definition, which should be easier to follow when looking at the examples provided in Figure 1.

[1] The notation $\mathfrak{Q}_{(1)}(\mathcal{P})$ might be more appropriate, since $\mathfrak{Q}_{(1)}$ is depending on $\mathcal{P}$, but we preferred to stay with this simpler notation.

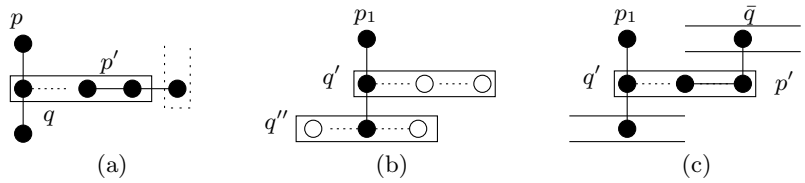


Fig. 2. The black vertices and solid edges indicate the P_2-packing $\mathcal{P}$. The polygons contain the P_2's of the packing $\mathcal{Q}$.

Definition 1. *1. We call* $q = q_1q_2q_3 \in \mathcal{Q}$ foldable *on* $p = p_1p_2p_3 \in \mathcal{P}$ *if, for* $q_2 \in V(p) \cap V(q)$*, we have* $p_s = q_2$*,* $s \in \{1, 2, 3\}$*, and either* $p_{s+1} \notin V(\mathcal{Q})$ *or* $p_{s-1} \notin V(\mathcal{Q})$*, see Figure 1(a).*

2. If q *is foldable on* p*, then substituting* q *by* $(q \setminus \{q_i\}) \cup \{p_{s\pm1}\}$ *with* $i \in \{1, 3\}$ *and* $q_i \notin V(p)$*, will be called* $(q_i, p_{s\pm1})$-folding*, see Figure 1(b).*

3. We call $q = q_1q_2q_3 \in \mathcal{Q}$ shiftable *with respect to* q_1 *(or* q_3*, resp.) on* $p = p_1p_2p_3 \in \mathcal{P}$ *if the following holds:* $q_1 \in V(p) \cap V(q)$ *(or* $q_3 \in V(p) \cap V(q)$*, resp.) and either* $p_{s+1} \notin V(\mathcal{Q})$ *or* $p_{s-1} \notin V(\mathcal{Q})$ *where* $p_s = q_1$ *(or* $p_s = q_3$*, resp.) and* $s \in \{1, 2, 3\}$*, see Figure 1(c).*

4. If q *is shiftable on* p *with respect to* $t \in \{q_1, q_3\}$*, then substituting* q *by* $q \setminus \{g\} \cup \{p_{s+1}\}$ *(or by* $q \setminus \{g\} \cup \{p_{s-1}\}$*, resp.),* $g \in \{q_1, q_3\} \setminus \{t\}$*, will be called* (g, p_{s+1})-shifting *(or* (g, p_{s-1})-shifting*, resp.), see Figure 1(d).*

Lemma 2. *If* $q = q_1q_2q_3 \in \mathcal{Q}$ *with* $\mathcal{Q} \in \mathfrak{Q}_{(2)}$ *is shiftable on* $p \in \mathcal{P}$ *with respect to one of the endpoints* q_e*,* $(e \in \{1, 3\})$*, then there is some* $p' \in \mathcal{P}$ *with* $p' \neq p$ *with* $\{q_{e'}, q_2\} \in E(p')$ *(*$e' \in (\{1, 3\} \setminus \{e\})$*), see Figure 2(a).*

Proof. W.l.o.g., discuss $e = 1$. We examine the case where $V(p) \cap V(q) = \{q_1\}$ and, w.l.o.g., $p_{s+1} \notin V(\mathcal{Q})$. Now assume the contrary. Then by (q_3, p_{s+1})-shifting, we obtain a P_2-packing $\mathcal{Q}'$. Comparing $\mathcal{Q}$ and $\mathcal{Q}'$ with respect to property (1), $\mathcal{Q}'$ is no worse than $\mathcal{Q}$. But $\mathcal{Q}'$ improves on property 2, as we gain $\{p_s, p_{s+1}\}$ without losing an edge from $\mathcal{P}$. But this contradicts $\mathcal{Q} \in \mathfrak{Q}_{(2)}$. □

Lemma 3. *If* $\mathcal{Q} \in \mathfrak{Q}_{(2)}$*, then no* $q \in \mathcal{Q}$ *is foldable.*

Proof. Suppose some $q \in \mathcal{Q}$ is foldable on p and, w.l.o.g., $p_{s+1} \notin V(\mathcal{Q})$. Then by (q_1, p_{s+1})-folding q we could improve on property 2 (without weakening property 1), contradicting $\mathcal{Q} \in \mathfrak{Q}_{(2)}$. □

Suppose there is a path p with $|V(p) \cap V(\mathcal{Q})| = 2$. Then p shares exactly one vertex $p_{q'}, p_{q''}$ with paths $q', q'' \in \mathcal{Q}$ due to Corollary 2. In the following $p_{q'}$ and $p_{q''}$ will always refer to the two cut vertices of the paths $q', q'' \in \mathcal{Q}$ which cut a path p with $|V(p) \cap V(\mathcal{Q})| = 2$.

Lemma 4. *Let* $\mathcal{Q} \in \mathfrak{Q}_{(2)}$*. Consider* $p \in \mathcal{P}$ *with* $|V(p) \cap V(\mathcal{Q})| = 2$ *and neither* $p_{q'}$ *nor* $p_{q''}$ *are* $\mathcal{Q}$*-endpoints. Then one of* q', q'' *is foldable.*

Proof. Let $i, j \in \{1, 2, 3\}$ such that $p_{q'} = p_i$ and $p_{q''} = p_j$. Then for $f \in \{1, 2, 3\} \setminus \{i, j\}$, we have $p_f \notin V(\mathcal{Q})$. W.l.o.g., $\{p_i, p_f\} \in E(p)$. Then q' is (q'_1, p_f)-foldable. □

Corollary 3. *Let $\mathcal{Q} \in \mathfrak{Q}_{(2)}$ and $p \in \mathcal{P}$ with $|V(p) \cap V(\mathcal{Q})| = 2$. Then one of $p_{q'}, p_{q''}$ must be a $\mathcal{Q}$-endpoint.*

Proof. Assume the contrary. Then by using Lemmas 3 and 4 we derive a contradiction. □

We define $\mathcal{P}_i(\mathcal{Q}) := \{p \in \mathcal{P} \mid i = |p \cap V(\mathcal{Q})|\}$.
Now we are ready to prove the main theorem.

Proof. (of Theorem 4) Suppose there is a path $p \in \mathcal{P}$ with $|V(p) \cap V(\mathcal{Q})| < 3$. By Corollary 2 we must have $|V(p) \cap V(\mathcal{Q})| = 2$. By Corollary 3, w.l.o.g.. $p_{q'}$ is a $\mathcal{Q}$-endpoint. For $p_{q''}$ there are two possibilities:
a) $p_{q''}$ is also a $\mathcal{Q}$-endpoint. Let $\{p_f\} = V(p) \setminus \{p_{q'}, p_{q''}\}$. Then, w.l.o.g., $\{p_{q'}, p_f\} \in E(p)$. Therefore $p_{q'}$ is shiftable.
b) $p_{q''}$ is a $\mathcal{Q}$-midpoint.
Claim. $p_{q''} \neq p_2$: Suppose the contrary. Then w.l.o.g., $p_{q'} = p_1$ and thus q'' is foldable on p by a (q''_1, p_3)-folding. This contradicts Lemma 3. The claim follows. W.l.o.g., we assume $p_{q''} = p_1$, see Figure 2(b). Then it follows that $p_{q'} = p_2$, as otherwise a (q''_1, p_2)-folding would contradict Lemma 3 again. From $p_{q'} = p_2$ and $p_3 \notin V(\mathcal{Q})$ we can derive that also in this case $p_{q'}$ is shiftable.

We now examine for both cases the implications of the shiftability of $p_{q'}$. W.l.o.g., we suppose that $p_{q'} = q'_1$. Due to Lemma 2 there is a $p' \in \mathcal{P}$ with $\{q'_3, q'_2\} \in E(p')$. From Corollary 2, it follows that there must be a $\bar{q} \in \mathcal{Q} \setminus \{q'\}$ with $|V(p') \cap V(\bar{q})| = 1$, see Figure 2(c). Hence, $|V(p') \cap V(\mathcal{Q})| = 3$. Note that q' is the only path in $\mathcal{Q}$ with $|V(q') \cap V(p')| = 2$. Summarizing, we can say that for any $p \in \mathcal{P}$ with $|V(p) \cap V(\mathcal{Q})| = 2$ we find a distinct $p' \in \mathcal{P}$ (via q') such that $|V(p') \cap V(\mathcal{Q})| = 3$. So, there is a total injection γ from $\mathcal{P}_2(\mathcal{Q})$ to $\mathcal{P}_3(\mathcal{Q})$. From $|\mathcal{P}_2(\mathcal{Q}) \cup \mathcal{P}_3(\mathcal{Q})| = j$ and the existence of γ we derive $|\mathcal{P}_2(\mathcal{Q})| \leq 0.5j$. This implies $|V(\mathcal{P}) \cap V(\mathcal{Q})| = 2|\mathcal{P}_2(\mathcal{Q})| + 3|\mathcal{P}_3(\mathcal{Q})| \geq 2.5j$. □

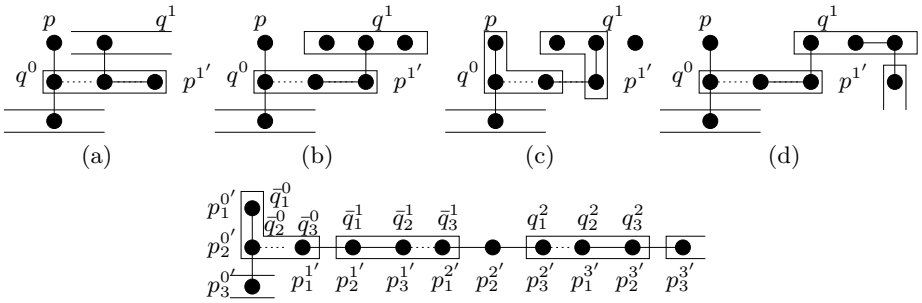
In a further step we will prove that also a considerable amount of old midpoints (vertices in $\mathcal{M}_\mathcal{P}$) will also be new midpoints (vertices in $\mathcal{M}_\mathcal{Q}$). For this reason we introduce a third priority. We only consider P_2-packings $\mathcal{Q} \in \mathfrak{Q}_{(2)}$ which maximize the following property (3) and subsume them in $\mathfrak{Q}_{(3)}$.

$$|\{q \in \mathcal{Q} \mid \exists p, p' \in \mathcal{P} : |V(p) \cap V(q)| = 1, p \in \mathcal{P}_2(\mathcal{Q}), |E(p') \cap E(q)| = 1\}| \quad (3)$$

Property (3) refers to maximizing situations depicted in Figure 3(a).

Lemma 5. *Let $\mathcal{Q} \in \mathfrak{Q}_{(3)}$ such that $|V(\mathcal{P}) \cap V(\mathcal{Q})| \leq (3 - \ell) \cdot j$ for some arbitrary natural number j and some number $\ell \in [0, 0.5]$. Then, $|\mathcal{M}_\mathcal{P} \cap \mathcal{M}_\mathcal{Q}| \geq \ell j$.*

Note that we can assume $0 \leq \ell \leq 0.5$ due to Theorem 4.



(e) The packing $\mathcal{Q}^2$.

Fig. 3.

Proof. We want to show the following relation: for every vertex $m \in V(\mathcal{Q})\backslash V(\mathcal{P})$ we find a unique midpoint $m' \in \mathcal{M}_\mathcal{P} \cap \mathcal{M}_\mathcal{Q}$.
Browsing the proof of Theorem 4 we find that there are ℓj paths p with $|V(p) \cap V(\mathcal{Q})| = 2$. For every such path p we can find a path $p' =: p^{1'}$ (via $q' =: q^0$) such that not only $|V(p^{1'}) \cap V(\mathcal{Q})| = 3$ but also $|E(p^{1'}) \cap E(q^0)| = 1$.
1) Let $\{e\} = E(p^{1'}) \cap E(q^0)$. Suppose $e = \{q_2^0, q_3^0\}$ with $q_2^0 = p_2^{1'}$ and $q_3^0 = p_1^{1'}$ (see Figure 3(a)) then $p_2^{1'} \in \mathcal{M}_\mathcal{P} \cap \mathcal{M}_\mathcal{Q}$ and the lemma follows for this case as $p_1^{1'}$ is distinct due to $|E(p^{1'}) \cap E(q^0)| = 1$.
2) The remaining case is $q_2^0 = p_1^{1'}$ and $q_3^0 = p_2^{1'}$. *a*) Suppose now there is $q^1 \in (\mathcal{Q} \setminus \{q^0\})$ with $|V(q^1) \cap V(p^{1'})| = 1$ (Cor. 2) and $V(q^1) \cap V(p^{1'}) \in \mathcal{M}_\mathcal{Q}$ (see Figure 3(b)). Then by substituting q by $\bar{q} := p_1 q_1^0 q_2^0$ (shifting) we obtain a packing $\mathcal{Q}^1$ with $|\mathcal{Q}^1| = |\mathcal{Q}|$ such that the path q^1 becomes foldable (see Figure 3(c)) which contradicts Lemma 3.
b) Hence we must have $V(q^1) \cap V(p^{1'}) = \{q_1^1\} = \{p_3^{1'}\}$ (see Figure 3(d)). Consider $\mathcal{Q}^1$: q^1 must be shiftable on $p^{1'}$ and hence there is $p^{2'} \in \mathcal{P}$ with $|E(p^{2'}) \cap E(q^1)| = 1$ and a $q^2 \in \mathcal{Q}$ with $|V(q^2) \cap V(p^{2'})| = 1$.
If we have $q_2^1 = p_2^{2'}$ and $q_3^1 = p_1^{2'}$ then $\mathcal{Q}^1$ improves property (3) with respect to $\mathcal{Q}$. Hence $\mathcal{Q} \notin \mathfrak{Q}_{(3)}$ which is a contradiction to the premise. Therefore we have $q_2^1 = p_1^{2'}$ and $q_3^1 = p_2^{2'}$. If $V(q^2) \cap V(p^{2'}) \in \mathcal{M}_\mathcal{Q}$ we derive a contradiction similar to case 2.*a*). Hence $V(q^2) \cap V(p^{2'}) = \{q_1^2\} = \{p_3^{2'}\}$.
By shifting q^1 (i.e., substituting q^1 by $\bar{q}^1 := p_2^{1'} q_1^1 q_2^1 = p_2^{1'} p_3^{1'} p_1^{2'}$; we call this *chain-shifting*) we obtain a packing $\mathcal{Q}^2$. By continued chain-shifting (i.e., substituting q^i by $\bar{q}^i = p_2^{i'} q_1^i q_3^i = p_2^{i'} p_3^{i'} p_1^{(i+1)'}$) we get a sequence of packings $\mathcal{Q}\mathcal{Q}^1\mathcal{Q}^2\ldots$ from $\mathfrak{Q}_{(3)}$ and paths $\bar{q}\bar{q}^1\bar{q}^2\ldots$ and $p^{0'}p^{1'}p^{2'}p^{3'}\ldots$. (where $p^{0'} := p$). We prove two claims to finally determine a contradiction in case 2.*b*).

Claim (1). For all $u \geq 1$ and $0 \leq s < u$ we have $\bar{q}^s \neq q^u$.

Proof (Claim (1)). Suppose the contrary that there are $u \geq 1$ and $0 \leq s < u$ with $\bar{q}^s = q^u$. Choose u as small as possible which means that q^u will be the first P_2 from the sequence which will be shifted for the second time.
For a packing $\mathcal{Q}^h$ and for any $k < h$ we have: $V(\bar{q}^k) \cap V(p^{k'}) = \{\bar{q}_1^k, \bar{q}_2^k\} =$

$\{p_2^{k'}, p_3^{k'}\}$ ($|E(\bar{q}^k) \cap E(p^{k'})| = 1$, resp.) and $V(\bar{q}^k) \cap V(p^{(k+1)'}) = \{\bar{q}_3^k\} = \{p_1^{(k+1)'}\}$. For $z \geq h$ we have $V(q^z) \cap V(p^{z'}) = \{q_1^z\} = \{p_3^{z'}\}$ and $V(q^z) \cap V(p^{(z+1)'}) = \{q_2^z, q_3^z\} = \{p_1^{(z+1)'}, p_2^{(z+1)'}\}$ ($|E(q^z) \cap E(p^{(z+1)'})| = 1$, resp.), see Figure 3(e) for the case $h = 2$.
Let us fix $\mathcal{Q}^u$. *a)* $s = u - 1$: We have $V(\bar{q}^{u-1}) \cap V(p^{u'}) = \{p_1^{u'}\}$ and $V(q^u) \cap V(p^{u'}) = \{p_3^{u'}\}$. Due to $\bar{q}^{u-1} = q^u$ this implies $p_1^{u'} = p_3^{u'}$, a contradiction.
b) $s < u-1$: Here we have $|V(\bar{q}^s) \cap V(p^{(s+1)'})| = 1$ but also $|V(q^u) \cap V(p^{u'})| = 1$. Due to $\bar{q}^s = q^u$ and the fact that q^u shares with exactly one path from $\mathcal{P}$ exactly one vertex, it follows $p^{(s+1)'} = p^{u'}$. In $\mathcal{Q}^s$ we must have $|V(q^s) \cap V(p^{(s+1)'})| = 2$ as q^s has not been chain-shifted yet. But due to the same reason we also must have $|V(q^{u-1}) \cap V(p^{(u)'})| = 2$. But as $p^{(s+1)'} = p^{u'}$ this means that $V(q^{u-1}) \cap V(q^s) \neq \emptyset$. Due to $q^s \neq q^{u-1}$ this contradicts the fact that $\mathcal{Q}^s$ is indeed a P_2-packing. □

Claim (2). All the elements in the sequence $\bar{q}^0 \dots \bar{q}^{u-1} q^u$ are pairwise different.

Proof (Caim (2)). We show the claim by induction on u. For $u = 1$ suffices *Claim (1)*. Now suppose for some u *Claim (2)* holds. Then also $\bar{q}^0 \dots \bar{q}^{u-1} \bar{q}^u$ are pairwise different as they also form a P_2-packing. Now by the preceding claim we know that $q^{u+1} \neq \bar{q}^s$ where $0 \leq s < u+1$ which closes the inductive step. □

We finally can deduce a contradiction in the case 2.*b*). Take the packing $\mathcal{Q}^{n+1}$ where due *Claim (2)* the P_2's $\bar{q}^0 \dots \bar{q}^n q^{n+1}$ form a packing in G and hence contain $3(n+1)$ different vertices. This contradicts the fact that $|V(G)| = n$. □

4 The Algorithm

The overall algorithm we propose has two core building blocks: kernelization and exhaustive search (based on the combinatorial properties derived in the previous section). Actually, both approaches are intertwined in a way that makes it necessary to first present, in very concise way, the kernelization results obtained in other papers. Moreover, it enables us to present the overall algorithm in a self-contained fashion.

4.1 Kernel Results

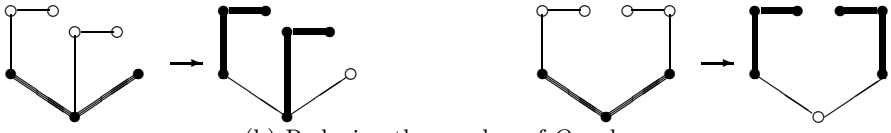
Theorem 3 was obtained by optimizing the use of fat and double crowns through local improvements, called **Rule 1** (see Figure 4(a)) and **Rule 2** (see Figure 4(b)). Given a maximal packing $\mathcal{P}$ then $R := V \setminus V(\mathcal{P})$ consists of single vertices (found in the set Q_0) and single edges (found in the set Q_1).

As we will use some properties of the kernel in the analysis of forthcoming algorithm we give a top level description of it.

A *double crown decomposition* of a graph G is a decomposition (H, C, R) of the vertices in G such that

1. H (the head) separates C and R;

(a) Reducing the number of Q_0-vertices



(b) Reducing the number of Q_1-edges

Fig. 4. Rule 1 and **Rule 2**. We use solid circles and thick lines for vertices and edges, respectively, in the P_2-packing $\mathcal{P}$, and hollow circles and thin lines for vertices and edges not in $\mathcal{P}$. Hence, two hollow circles linked by a thin line represents a Q_1-edge.

2. $C = C_0 \cup C' \cup C''$ (the crown) is an independent set such that $|C'| = |H|$, $|C''| = |H|$, and there exist a perfect matching between C' and H, and a perfect matching between C'' and H.

A *fat crown decomposition* of a graph G is a decomposition (H, C, R) of the vertices in G such that

1. H (the head) separates C and R;
2. the induced subgraph $G(C)$ is a collection of pairwise disjoint K_2's;
3. there is a perfect matching M between H and a subset of vertices in C such that each connected component in $G(C)$ has at most one vertex in M.

Lemma 6 ([29]). *A graph G with a double crown (fat crown, resp.) (H, C, R) has a P_2-packing of size k if and only if the graph $G - H - C$ has a P_2-packing of size $k - |H|$.*

Lemma 6 is applied in Alg. 1 in lines 7 and 9. Here the lines 1-12 correspond to the kernelization algorithm of [4, 31], except that in the 'else'-part of line 12 we return the current kernelized graph G_{ker} (and omit the exponential part). G_{ker} consists of a maximal P_2-packing $\mathcal{P}$ of size $j \leq k-1$ and of R. Now [4, 31] proved the following:

Lemma 7 ([4, 31]). *In line 9 of Alg. 1 when we return G_{ker} we have:*
1. $|Q_0| \leq 2k$ *and* 2. $|Q_1| \leq k$.

We mention that if the bounds given in Lemma 7 are not met then it is proven that a fat or double crown can be applied (see lines 6-7 and 8-9 in Alg. 1). Putting things together we have that $|V(G_{ker})| \leq 7k$.

4.2 Combining Kernel and Combinatorial Properties

Basically, Algorithm 1 inductively improves over its current packing $\mathcal{P}$ by using exhaustive search as detailed in Algorithm 2 and justified in the preceding section. To obtain the claimed run time bounds, it actually first kernelizes and then tries to augment its solution by exhaustive search.

Algorithm 1 An Algorithm for P_2-PACKING.

1: $\mathcal{P} = \emptyset$.
2: Greedily augment $\mathcal{P}$ by one more P_2.
3: **If** step 2 succeeded **then Goto** 2.
4: Apply **Rule 1**. If it applied **Goto** 2.
5: Apply **Rule 2**. If it applied **Goto** 2.
6: **if** $|Q_0| \geq 2|\mathcal{P}|$ **then**
7: construct a double crown (H, C, R); $k \leftarrow k - |H|$; $G \leftarrow G - H - C$, **Goto** 1;
8: **else if** $|Q_1| \geq |\mathcal{P}|$ **then**
9: construct a fat crown (H, C, R); $k \leftarrow k - |H|$; $G \leftarrow G - H - C$; **Goto** 1;
10: **else if** $k \leq 0$ **then**
11: **return** YES
12: **else**
13: Try to construct a P_2-packing $\mathcal{P}'$ from $\mathcal{P}$ with $|\mathcal{P}| + 1 = |\mathcal{P}'|$ using Algorithm 2.
14: **if** Step 13 failed **then**
15: **return** NO.
16: **else**
17: $\mathcal{P} \leftarrow \mathcal{P}'$.
18: **Goto** 2.

Algorithm 2 An Algorithm for augmenting a maximal P_2-packing $\mathcal{P}$.

1: $j \leftarrow |\mathcal{P}|$.
2: **for** ℓ=0 to $0.25j$ **do**
3: **for all** $S_i \subseteq V(\mathcal{P})$, $S_o \subseteq V \setminus V(\mathcal{P})$ with $|S_i| = (j+1) - \ell$ and $|S_o| = \ell$ **do**
4: Try to construct a P_2-packing $\mathcal{P}'$ with $S_i \cup S_o$ as midpoints.
5: **if** Step 4 succeeded **then**
6: **return** $\mathcal{P}'$.
7: **for** $\bar{\ell} = 0.25j$ to $0.5j + 3$ **do**
8: **for** $r = \bar{\ell}$ to $(j+1) - \bar{\ell}$ **do**
9: **for all** $B_o \subseteq V \setminus V(\mathcal{P})$, $X \subseteq \mathcal{M}_{\mathcal{P}}$, $B_i \subseteq (V(\mathcal{P}) \setminus \mathcal{M}_{\mathcal{P}})$ with $|B_o| = \bar{\ell}$, $|X| = r$ and $|B_i| = (j+1) - \bar{\ell} - r$ **do**
10: Try to construct a P_2-packing $\mathcal{P}'$ with $B_i \cup B_o \cup X$ as midpoints.
11: **if** Step 9 succeeded **then**
12: **return** $\mathcal{P}'$.
13: **return** failure.

We are going to discuss three main aspects of Algorithm 2: (1) how matching techniques can be used in the WIN-WIN-approach, (2) why the algorithm is yielding a correct solution, and (3) how the run time is estimated.

4.3 Used Matching Techniques

We would like to point out the following fact about P_2-packings. If a graph has a P_2-packing $\mathcal{P} = \{p^1, \ldots, p^k\}$, then it suffices to know the set of midpoints $\mathcal{M}_{\mathcal{P}} = \{p_2^1, \ldots, p_2^k\}$ to construct a P_2-packing of size k (which is possibly $\mathcal{P}$) in polynomial time. This fact was discovered by E. Prieto and C. Sloper [29]

and basically can be achieved by bipartite matching techniques. Details on the mentioned matching technique can be found in the following proposition.

Proposition 1. *Let the vertex set $\mathcal{M} = \{m_1, \ldots, m_j\}$ contain all the midpoints of some P_2-packing $\mathcal{P}$ in a graph $G(V, E)$. Then we can construct a P_2-packing $\mathcal{P}'$ of size j in polynomial time.*

Proof. Use the following algorithm:

- Find a maximum matching M in the auxiliary bipartite graph $G' = (V', E')$, where $V' = A \cup B$ is the bipartition with $A = \mathcal{M} \times \{1, 2\}$ and $B = V \setminus \mathcal{M}$, $E' = \{\{(u, i), w\} \mid 1 \leq i \leq 2, (u, i) \in A, w \in B, \{u, w\} \in E\}$.
- If all elements of A are matched in M, then we have found a packing $\mathcal{P}'$ of G as follows: $\mathcal{P}' = \{(x, y, z) \mid \{\{(y, 1), x\}, \{(y, 2), z\}\} \subseteq M\}$.

Note that $M_{\mathcal{P}} = \{\{\{(p_2, 1), p_3\}, \{(p_2, 2), p_1\}\} \mid p_1p_2p_3 \in \mathcal{P}\}$ matches A into B in G'. Thus, $\mathcal{P}'$ must exist and is of size j. □

4.4 Correctness

The correctness of the kernelization part is shown in [4, 31]. If a P_2-packing $\mathcal{P}'$ with $|\mathcal{P}'| = j + 1$ exists, we can partition the midpoints $\mathcal{M}_{\mathcal{P}'}$ in a part which lies within $V(\mathcal{P})$ and one which lies outside. We call them $\mathcal{M}^i_{\mathcal{P}'} := \mathcal{M}_{\mathcal{P}'} \cap V(\mathcal{P})$ and $\mathcal{M}^o_{\mathcal{P}'} := \mathcal{M}_{\mathcal{P}'} \cap O$, respectively with $O := V(\mathcal{P}') \setminus V(\mathcal{P})$. Theorem 4 yields $|O| \leq 0.5j + 3$ and thus $|\mathcal{M}^o_{\mathcal{P}'}| \leq 0.5j + 3$.
Basically, we can find an integer ℓ with $0 \leq \ell \leq 0.5j + 3$ such that $|\mathcal{M}^i_{\mathcal{P}'}| = (j + 1) - \ell$ and $|\mathcal{M}^o_{\mathcal{P}'}| = \ell$. In step 2 of Alg. 2, we run through every such ℓ until we reach $0.25j$. For any choice of ℓ, in step 3 of Alg. 2, we cycle through all possibilities of choosing sets $S_i \subseteq V(\mathcal{P})$ and $S_o \subseteq V \setminus V(\mathcal{P})$ such that $|S_i| = (j + 1) - \ell$ and $|S_o| = \ell$. Here S_i and S_o are candidates for $\mathcal{M}^i_{\mathcal{P}'}$ and $\mathcal{M}^o_{\mathcal{P}'}$, respectively. For any choice of S_i and S_o, we try to construct a P_2-packing. If we succeed once, we can return the desired larger P_2-packing.
Otherwise, we reach the point where $\ell = 0.25j$. Due to Lemma 5 we know that if $|\mathcal{M}^o_{\mathcal{P}'}| = \bar{\ell}$ then also $|\mathcal{M}_{\mathcal{P}} \cap \mathcal{M}_{\mathcal{P}'}| =: r \geq \bar{\ell}$. Then due to Theorem 4 we can find an $0.25j \leq \bar{\ell} \leq 0.5j + 3$ and $\bar{\ell} \leq r \leq (j + 1) - \bar{\ell}$ such that $|\mathcal{M}^o_{\mathcal{P}'}| = \bar{\ell}$, $|\mathcal{M}_{\mathcal{P}} \cap \mathcal{M}_{\mathcal{P}'}| = r$ and $|\mathcal{M}^i_{\mathcal{P}'} \setminus \mathcal{M}_{\mathcal{P}}| = (j + 1) - \bar{\ell} - r$. In step 8 of Alg. 2 we cycle through all candidate sets B_o, B_i for $\mathcal{M}^o_{\mathcal{P}'}, (\mathcal{M}^i_{\mathcal{P}'} \setminus \mathcal{M}_{\mathcal{P}})$ but also through X which is a candidate for $\mathcal{M}_{\mathcal{P}} \cap \mathcal{M}_{\mathcal{P}'}$. As we are already looking for $\bar{\ell} + r$ midpoints in $\mathcal{M}^o_{\mathcal{P}'} \cup (\mathcal{M}_{\mathcal{P}} \cap \mathcal{M}_{\mathcal{P}'})$ we only have to find $(j + 1) - \bar{\ell} - r$ midpoints in $(V(\mathcal{P}) \setminus \mathcal{M}_{\mathcal{P}}) \supseteq (\mathcal{M}^i_{\mathcal{P}'} \setminus \mathcal{M}_{\mathcal{P}})$.

4.5 Running Time

The only exponential run time contribution comes from the **for**-loops in Alg. 2. For any ℓ we execute step 3 of Alg. 2 at most $\binom{3j}{(j+1)-\ell}\binom{qj}{\ell} \in \mathcal{O}\left(\binom{3j}{j-\ell}\binom{qj}{\ell}\right)$ times, since $|V(\mathcal{P})| = 3j$ and $|V \setminus V(\mathcal{P})| = qj$ $(0 \leq q \leq 4)$ due to Theorem 3. Likewise, $\mathcal{O}\left(\binom{qj}{\bar{\ell}}\binom{j}{r}\binom{2j}{j-\bar{\ell}-r}\right)$ upper-bounds step 9 of Alg. 2 as we have $|V \setminus V(\mathcal{P})| = qj$, $|\mathcal{M}_{\mathcal{P}}| = j$, and $|V(\mathcal{P}) \setminus \mathcal{M}_{\mathcal{P}}| = 2j$.

Lemma 8. *Let* $A(q,z)[j] := \binom{3j}{j-zj}\binom{qj}{zj}$ *with* $0 \le q \le 4$, $0 \le z \le 0.25$ *and* $z \le q$. *Then, for all admitted values of* q, z, *we have* $A(q,z)[j] \in \mathcal{O}(14.67^j)$.

Proof. We are going to show that $\binom{3j}{j-b}\binom{4j}{b} < \binom{3j}{j-(b+1)}\binom{4j}{b+1}$ for $0 \le b \le 0.5j-1$. We have: $\binom{3j}{j-(b+1)}\binom{4j}{b+1} - \binom{3j}{j-b}\binom{4j}{b} = \frac{(3j)!(4j)!((j-b)(4j-b)-(2j+b+1)(b+1))}{(j-b)!(2j+b+1)!(b+1)!(4j-b)!}$.
It follows that

$$\begin{aligned}((j-b)(4j-b) &- (2j+b+1)(b+1))\\ &= 4j^2 - jb7 - 2j - 2b - 1\\ &\overset{b=0.5j-1}{\ge} 0.5j^2 + 4j + 1 > 0\end{aligned}$$

Hence, $A(q,z)[j] \le \binom{3j}{j-zj}\binom{4j}{zj} \le \binom{3j}{0.75j}\binom{4j}{0.25j} \in \mathcal{O}(13.77^j)$. □

Lemma 9. *Let* $B(q,z,s)[j] := \binom{qj}{zj}\binom{j}{sj}\binom{2j}{j-(z+s)j}$ *with* $0 \le q \le 4$, $0.25 \le z \le 0.5$, $z \le s \le 1-z$ *and* $z \le q$. *Then, for all admitted values of* q, z, s, *we have* $B(q,z,s) \in \mathcal{O}(14.67^j)$.

Proof. 1. We first show $B(q,z,z)[j] \ge B(q,z,s)[j]$. If we are able to prove $w := \binom{j}{b}\binom{2j}{j-zj-b} - \binom{j}{b+1}\binom{2j}{j-zj-(b+1)} > 0$ for any given $0.25 \le z \le 0.5$ and any integer $zj \le b \le (1-z)j-1$ we are done. We have

$$w = \frac{j!(2j)!(\overbrace{(b+1)(j+zj+b+1) - (j-b)(j-zj-b)}^{w'})}{(b+1)!(j-b)!(j-zj-b)!(j+zj+b+1)!}.$$

$$\begin{aligned}\text{It follows } w' &= z(j^2+j) + 3jb - j^2 + j + 2b + 1\\ &\overset{b=zj}{\ge} z(4j^2+3j) - j^2 + j + 1 \overset{z\ge 0.25}{\ge} \frac{7}{4}j + 1 > 0.\end{aligned}$$

2. Secondly, we show $B(4,\tau,\tau)[j] \ge B(4,z,z)[j]$ with $0.25 \le z \le 0.5$ and $\tau = 0.327528803767482$. Due to the relation $\binom{qj}{pj} \in \mathcal{O}\left(\left(\frac{q}{p}\right)^{pj}\left(\frac{q}{q-p}\right)^{(q-p)j}\right)$ (see Fernau [11]) $B(4,z,z)[j]$ is asymptotically upper-bounded by the expression:

$$\left(\frac{\overbrace{\phantom{\left(\frac{4}{z}\right)^z\left(\frac{4}{4-z}\right)^{(4-z)}\left(\frac{1}{z}\right)^z\left(\frac{1}{1-z}\right)^{(1-z)}\left(\frac{2}{1-2z}\right)^{(1-2z)}\left(\frac{2}{1+2z}\right)^{(1+2z)}}}^{=:\lambda(z)}}{\left(\frac{4}{z}\right)^z\left(\frac{4}{4-z}\right)^{(4-z)}\left(\frac{1}{z}\right)^z\left(\frac{1}{1-z}\right)^{(1-z)}\left(\frac{2}{1-2z}\right)^{(1-2z)}\left(\frac{2}{1+2z}\right)^{(1+2z)}}\right)^j$$

By taking the derivative of $\lambda(z)$ and calculating its zeros we find that $\lambda(z)$ is maximum at the point τ. Here we made use of a computer algebra system. So, the result is correct modulo standard numerical errors.

Using subitems 1. and 2. we conclude: $B(q,z,s)[j] \le B(q,z,z)[j] \le B(4,z,z)[j] \le B(4,\tau,\tau)[j] \in \mathcal{O}(14.67^j)$. □

Theorem 5. P_2-PACKING *can be solved in time* $\mathcal{O}(2.448^{3k}k^{5.5} + k^2n^2)$.

Proof. In lines 4 and 10 of Alg. 2, we solve a bipartite matching problem which takes $\mathcal{O}(j^{2.5})$ time due to [18]. By Lemmas 8 and 9 the run times of steps 3 and 9 of Alg. 2 are both upper-bounded by $\mathcal{O}^*(14.67^j)$. Taking also steps 2, 7 and 8 into account, which each need less than j steps, Alg. 2 alone consumes $\mathcal{O}(2.448^{3k}k^{4.5})$ time.

Considering Alg. 1, we are able to construct a fat or double crown (C, H, R) such that C consists either of Q_1-edges or Q_0-vertices (steps 7 and 9 in Alg. 1) in linear time (see [29]). Both crown detections together will only be applied k times and the total amount of time is therefore $\mathcal{O}(kn)$ for all invoked crown reductions.

Rule 2 can only be applied k times and checking whether it applies takes $\mathcal{O}(kn)$ steps. Greedy augmentation can be done in n steps in the following manner: If an unpacked vertex v has two unpacked neighbors, we immediately have a new P_2. If v has only one unpacked neighbor u, check whether u has a third unpacked neighbor. If v has no unpacked neighbor, then delete v.

The most difficult part is to estimate the time for greedy augmentation (line 2) and **Rule 1**. First **Rule 1** can only be applied $n/2$ times repeatedly as every time two Q_0-vertices disappear, see [4]. We will divide the executions of line 2 into two parts: the actual augmentations by one P_2 (*aug*), which can only happen k times, and the non-augmentations (*no-aug*).

We claim that after at most $n/2 + 1$ no-augs without disruption of augs, we either decrement the parameter k or $\mathcal{P}$ will be augmented. Note that after a no-aug we immediately check if **Rule 1** applies. If it does not apply, then either **Rule 2**, a double or fat crown or the invocation of Alg. 2 applies and the claim follows. If it applies, we jump back to line 2. But after $n/2 + 1$ alternations between no-augs and **Rule 1**, **Rule 1** can not apply anymore and the claim follows. Hence, the time spent for no-augs is $\mathcal{O}(kn^2)$. Similarly we see that **Rule 1** takes a total of $\mathcal{O}(k^2n^2)$ steps. As Alg. 2 is invoked at most k times we obtain an overall run time of $\mathcal{O}(2.448^{3k}k^{5.5} + kn + k^2n + kn^2 + k^2n^2)$. □

Notice the asymptotic speed-up we achieve by changing the strategy (WIN-WIN). If we would skip the search for the old midpoints which are also new midpoints, we would have to count ℓ up to $0.5j + 3$ in step 3. Then, we had that $\binom{3j}{0.5j}\binom{4j}{0.5j} \in \mathcal{O}(17.44^j)$ (for j up to k), which is also not a big improvement compared to a brute force search for the midpoints on the $7k$-kernel. Namely, this would take $\mathcal{O}^*(17.66^k)$ steps.

We also like to point out that Lemma 5 is crucial to obtain the run time. Namely, without this result we would not have a lower bound $|\mathcal{M}_{\mathcal{P}} \cap \mathcal{M}_{\mathcal{P}'}| \geq |V(\mathcal{P}') \cap V(\mathcal{P})| - 3$. Hence, we would have to take into account expressions like $B(4, z, r)[j]$ where $r < z$. For example, if $z = 0.5$ and $r = 0.17$, then $B(4, z, r)[j] \in \mathcal{O}(17.44^j)$.

5 Lower Bound & Subcubic Graphs

The main topic of this section is to provide examples of families of (reduced) graphs for which the obtained reusability results are optimal. Some thoughts on degree-bounded graphs in a first subsection complement these findings.

5.1 Subcubic Graphs

A graph is called *subcubic* if the degree of any vertex is no greater than three. Kosowski et *al.* [21] obtained a linear time algorithm that for any subcubic graph outputs a P_2-packing of size at least $\frac{n}{5}$. For a subcubic graphs without vertices of degree one (so called $(2,3)$-graphs) this results can be even improved to $\frac{n}{4}$. We can use this results to obtain better results for these graph classes.

Theorem 6. P_2-PACKING *on subcubic graphs can be solved in time* $\mathcal{O}^*(1.4534^{3k})$ *and on* $(2,3)$*-graphs in* $\mathcal{O}^*(1.2318^{3k})$.

Proof. Subcubic graphs: We first compute a packing $\mathcal{P}$ containing $\frac{3}{5}n$ vertices as mentioned above. If $k \leq \frac{3}{5}n$ we are done. Otherwise we have $n < \frac{5}{3}k$. Finding the k midpoints then takes $\mathcal{O}(\binom{\frac{5}{3}k}{k}) \subseteq \mathcal{O}(1.4534^{3k})$ steps.
$(2,3)$*-graphs:* We now are able to compute a packing with at least $\frac{3}{4}n$ vertices. Similarly we can compute a packing of size k if one exists in $\mathcal{O}(\binom{\frac{4}{3}k}{k}) \subseteq \mathcal{O}(1.2839^{3k})$. □

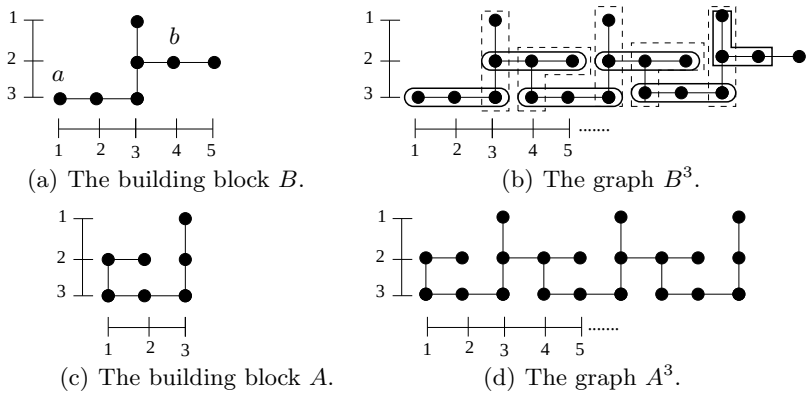
5.2 Lower Bound for Vertex Reusability

In this section, we show that Theorem 4 is indeed sharp for general graphs and a general maximal P_2-packing $\mathcal{P}$. We will expose a graph family B^ℓ such that any P_2-packing of size $j+1$, where $j = |\mathcal{P}|$, will asymptotically recycle at most $2.5j$ out of the $3j$ vertices in $V(\mathcal{P})$.

A Lower Bound Example The graph family is defined the following way: $B^1 = B$, where B is the graph in Figure 5(a) and the vertices are indexed by the indicated grid. Here the vertical axis corresponds to the first entry and the horizontal axis to the second entry. The graph B^ℓ will emerge if we take B^1 and $B^{\ell-1}$ and connect vertex $(3,1)$ of $B^{\ell-1}$ with vertex $(2,4)$ of B^1, see Figure 5(b) for an example of B^3. The vertices of B^ℓ will be indexed according to the grid in Figure 5(b). The parts of B^ℓ which correspond to B will be indexed from the left to the right as B^ℓ_i.

We call a *pending* P_2 an induced P_2 which is attached to the rest of the graph G by exactly one edge. In Figure 5(a) the three left most vertices form a pending P_2. Note that a pending P_2 is a double crown if it is attached to G via the midpoint and a fat crown if it is attached via one of its endpoints. Therefore, the next reduction rule is sound:

Pending: Let p be a pending P_2. Then delete P_2 and decrease k by one.



(a) The building block B. (b) The graph B^3.

(c) The building block A. (d) The graph A^3.

Fig. 5. The graphs A^ℓ and B^ℓ and the non-optimal P_2-packing $\mathcal{P}$ (dotted rectangles in Figure 5(b)) and the optimal P_2-packing $\tilde{\mathcal{P}}$ (solid rectangles in Figure 5(b)) of B^3 are depicted.

Lemma 10. *Let $\mathcal{P}'$ be a maximum P_2-packing of B^ℓ. Then $|\mathcal{P}'| = 2\ell$.*

Proof. We proceed by induction on ℓ. For $\ell = 1$ the claim is true as two **pending** applications with respect to B^1 show. Let $\ell > 1$, then pick B^ℓ_1. The reduction rule **pending** applies to the P_2 induced by $(3,1),(3,2),(3,3)$ and afterwards also to the P_2 $((1,3),(2,3),(2,4))$. As these two P_2 will be deleted the remaining graph is a $B^{\ell-1}$ and an isolated K_1. Now the proof follows by induction. □

We introduce another graph family A^ℓ. It is also defined recursively: $A^1 = A$, see Figure 5(c) and the vertices are indexed by indicated grid. A^ℓ is generated by taking A^1 and $A^{\ell-1}$ and connecting the vertex $(2,3)$ of A^1 with the vertex $(2,1)$ of $A^{\ell-1}$, see Figure 5(d) for an example of A^3. The vertices of A^ℓ will be indicated according to the grid in Figure 5(d). The parts of A^ℓ which correspond to A will be indexed from the left to the right as A^ℓ_i. We prove now the next Lemma as it is used as an intermediate step.

Lemma 11. *Let $\mathcal{P}'$ be a maximum P_2-packing of A^ℓ. Then $|\mathcal{P}'| = 2\ell$.*

Proof. We proceed by induction on ℓ. For $\ell = 1$ the claim is true as two **pending** applications with respect to A^1 show. Let $\ell > 1$, then pick A^ℓ_1. The reduction rule **pending** applies to the P_2 induced by $(2,2),(2,1),(3,1)$ and afterwards also to the P_2 $((3,2),(3,3),(2,3))$. As these two P_2 will be deleted the remaining graph is a $A^{\ell-1}$ and a K_1. Now the proof follows by induction. □

Theorem 7. *For the graph family B^ℓ defined above, there is a P_2-packing $\mathcal{P}$ of size $2\ell-1$ such that for every P_2-packing $\mathcal{P}'$ of size 2ℓ we have $|V(\mathcal{P}) \cap V(\mathcal{P}')| \leq 5\ell - 2$.*

Proof. First we define $\mathcal{P} := \tau_1 \cup \tau_2$ where $\tau_1 := \{((3,3+3i),(2,3+3i),(1,3+3i)) \mid 0 \leq i \leq \ell-1\}$ and $\tau_2 := \{((3,4+3j),(2,4+3j),(2,5+3j)) \mid 0 \leq j \leq \ell-2\}$, see

Figure 5(b) where the elements of $\mathcal{P}$ are framed by dotted rectangles in B^3.
Let $\tilde{\mathcal{P}} := \tilde{\tau}_1 \cup \tilde{\tau}_2 \cup \alpha$ where $\tilde{\tau}_1 := \{((3,1+3i),(3,2+3i),(3,3+3i)) \mid 0 \leq i \leq \ell-1\}$, $\tilde{\tau}_2 := \{((2,3+3j),(2,4+3j),(2,5+3j)) \mid 0 \leq j \leq \ell-2\}$ and $\alpha := \{((1,3\ell),(2,3\ell),(2,1+3\ell))\}$, see Figure 5(b) where the elements of $\tilde{\mathcal{P}}$ are framed by solid rectangles in B^3.
Observe that $\tilde{\mathcal{P}}$ exactly reuses $5\ell-2$ out of the $6\ell-3$ vertices of $\mathcal{P}$. The vertices of the P_2's in τ_2 will appear entirely in $V(\tilde{\mathcal{P}})$. Of each P_2 in τ_1 exactly two vertices are reused by $\tilde{\mathcal{P}}$ except for the one to the very right. Here all vertices are recycled. Summing up $3(\ell-1)+2(\ell-1)+3 = 5\ell-2$ vertices are recycled.

Claim. Let $\mathcal{P}'$ be an P_2-packing of size greater than $2\ell-1$ for B^ℓ such that $|V(\mathcal{P}) \cap V(\mathcal{P}')|$ is maximum. Then $|V(\mathcal{P}) \cap V(\mathcal{P}')| = |V(\mathcal{P}) \cap V(\tilde{\mathcal{P}})|$.

Proof. We must have $|\mathcal{P}'| = 2\ell$ due to Lemma 10. Again we use induction on ℓ. if $\ell = 1$ then $\tilde{\mathcal{P}} = \{((3,1),(3,2),(3,3)),((1,3),(2,3),(2,4))\}$ and $|V(\mathcal{P}) \cap V(\tilde{\mathcal{P}})| = 3$ which is optimal and hence shows the lemma for $\ell = 1$.
Let $\ell > 1$. Consider the set $L := \{(3,1),(3,2),(3,3)\}$. We must have $L \cap V(\mathcal{P}') \neq \emptyset$ as $\mathcal{P}'$ is maximal. We will now distinguish cases with respect to the expression $L \cap V(\mathcal{P}')$. It can be easily seen that the case distinction is indeed complete.

$L \cap V(\mathcal{P}') = \{(3,3)\}$: *a)* If $h := ((3,3),(2,3),(1,3)) \in \mathcal{P}'$, then $G[V \setminus V(h)]$ consists of an $A^{\ell-1}$ and an isolated K_2. Due to Lemma 11, any maximum P_2-packing for $A^{\ell-1}$ has size $2(\ell-1)$. Hence, $|\mathcal{P}' \setminus \{p\}| = |\{p \in \mathcal{P}' \mid V(p) \subseteq V(A^{\ell-1})\}| \leq 2(\ell-1)$. Therefore, $|\mathcal{P}'| \leq 2\ell-1$, a contradiction to the optimality of $\mathcal{P}'$.
b) If $g := ((3,3),(2,3),(2,4)) \in \mathcal{P}'$, then $G[V \setminus V(g)]$ consists of two isolated K_1's, a isolated K_2's and a $B^{\ell-1}$. With Lemma 10 (similarly to subitem *a*)), it follows that $|\mathcal{P}'| \leq 2\ell-1$, which is again a contradiction.

$L \cap V(\mathcal{P}') = \{(3,2),(3,3)\}$: Then, $f := ((3,2),(3,3),(2,3)) \in \mathcal{P}'$. $G[V \setminus V(f)]$ consists of two isolated K_1's and an $A^{\ell-1}$. With Lemma 11 (similarly to subitem *a*) in the previous item), it follows that $|\mathcal{P}'| \leq 2\ell-1$, a contradiction.

$L \cap V(\mathcal{P}') = \{(3,1),(3,2),(3,3)\}$: Then $d := ((3,1),(3,2),(3,3)) \in \mathcal{P}'$. Now consider the set $Q =: \{(1,3),(2,3),(2,4),(2,5)\}$. Due to the topology of B^ℓ, at most one P_2 from $\mathcal{P}'$ can share a vertex with Q. As $\mathcal{P}'$ is maximal, there must be exactly one P_2, say $p \in \mathcal{P}'$, that shares a vertex with Q.
Let $\hat{\mathcal{P}} = (\mathcal{P}' \setminus \{p\}) \cup c$ where $c := \{((2,3),(2,4),(2,5))\}$. As we have $Q \subseteq V(\mathcal{P})$ it follows that $|V(\mathcal{P}) \cap V(\mathcal{P}')| = |V(\mathcal{P}) \cap V(\hat{\mathcal{P}})|$. Now, $G[V \setminus (V(d) \cup V(c))]$ is a $B^{\ell-1}$ and an isolated K_1. Let $\tilde{\mathcal{P}}_{\ell-1}$ contain only those P_2's of $\tilde{\mathcal{P}}$ which are entirely contained in $B^{\ell-1}$. Let $\hat{\mathcal{P}}_{\ell-1}$ be defined the same way with respect to $\hat{P}$.
By induction, $|V(\mathcal{P}) \cap V(\tilde{\mathcal{P}}_{\ell-1})| \geq |V(\mathcal{P}) \cap V(\hat{\mathcal{P}}_{\ell-1})|$. As we have $\tilde{\mathcal{P}} = \tilde{\mathcal{P}}_{\ell-1} \cup \{d,c\}$ and $\hat{P} = \hat{P}_{\ell-1} \cup \{d,c\}$ we see that $|V(\mathcal{P}) \cap V(\tilde{\mathcal{P}})| \geq |V(\mathcal{P}) \cap V(\hat{\tilde{\mathcal{P}}})| = |V(\mathcal{P}) \cap V(\mathcal{P}')|$.

□

From the previous claim follows that no P_2-packing of size $j+1$ can reuse more vertices of $\mathcal{P}$ than $\tilde{\mathcal{P}}$. This proves the theorem finally. □

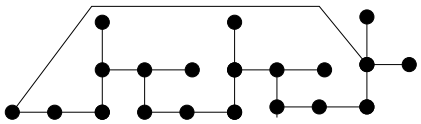


Fig. 6. The graph C^3.

From Theorem 7 we see that for any P_2-packing $\bar{\mathcal{P}}$ of size $j+1$ we have $\frac{|V(\mathcal{P})|}{|V(\mathcal{P}) \cap V(\bar{\mathcal{P}})|} \leq \frac{6\ell-3}{5\ell-2}$ and therefore $\lim_{\ell\to\infty} \frac{6\ell-3}{5\ell-2} = \frac{6}{5} = \frac{3j}{2.5j}$.
We see that Theorem 4 is indeed sharp. We point out that our example is also irreducible with respect to the $7k$-kernel. With respect to B^ℓ and $\mathcal{P}$ we have $|Q_0| = \ell - 1$. But as $|\mathcal{P}| = 2\ell - 1$ the double crown reduction rule does not apply (line 6 of Alg. 1). The same is true for fat crowns as $|Q_1| = 2$ (line 8 of Alg. 1). A serious drawback is that the **pending** reduction rule suffices to solve P_2-PACKING on B^ℓ. Thus, if we enrich the kernelization part by **pending** such a worst case instance never could show up in Alg. 2. In the last section we will give a slightly modified graph family which is also irreducible with respect to **pending**.

An Irreducible Lower Bound Example Let $C^\ell := G(V(B^\ell) \setminus \{(2, 2+3\ell)\}, (E(B^\ell) \setminus \{\{(2, 1+3\ell), (2, 2+3\ell)\}\}) \cup \{\{(3, 1), (2, 3\ell)\}\})$, thus we simply delete the vertex $(2, 2+3\ell)$ and add the edge $\{(3, 1), (2, 3\ell)\}$ to B^ℓ, see Figure 6. Due to this modification **pending** cannot be applied to C^ℓ. Nevertheless, we can find the packings $\mathcal{P}$ and $\tilde{\mathcal{P}}$ in C^ℓ.

Theorem 8. *Given C^ℓ and the P_2-packing $\mathcal{P}$ of size $2\ell-1$ for every P_2-packing $\mathcal{P}'$ of size 2ℓ we have the relation $|V(\mathcal{P}) \cap V(\mathcal{P}')| \leq 5\ell - 2$.*

Proof. Suppose there is P_2-packing $\mathcal{P}'$ of size $j+1$ with $|V(\mathcal{P}) \cap V(\mathcal{P}')| > 5\ell - 2$. Assume that there is no $p \in \mathcal{P}'$ such that $\{(3, 1), (2, 3\ell)\} \in E(p)$. Then $\mathcal{P}'$ is also a packing for B^ℓ. But this contradicts Theorem 7. Hence, there are two remaining possibilities:
a) $((3, 2), (3, 1), (2, 3\ell)) \in \mathcal{P}'$. Consider $G' := G[V \setminus \{(3, 2), (3, 1), (2, 3\ell)\}]$. One can show by induction that after $2\ell - 2$ applications of **pending**, the remaining graph consists of isolated K_1's and K_2's. We conclude $|\mathcal{P}'| = 2\ell - 1$, which is a contradiction to the premise.
b) $((3, 1), (2, 3\ell), (2, 1+3\ell)) \in \mathcal{P}'$ or $((3, 1), (2, 3\ell), (1, 3\ell)) \in \mathcal{P}'$: Similarly to *a)*, this would result in the contradictory consequence $|\mathcal{P}'| = 2\ell - 1$. □

By Theorem 8, we now have a graph family which is irreducible with respect to hitherto reduction rules, and this shows that Theorem 4 is (asymptotically) sharp. We would like to share the intuition that degree-one vertices which are attached to a vertex of degree greater one are crucial for our lower-bound example. This is in accordance with Section 5.1: The guaranteed P_2-packing for $(2, 3)$-graphs has greater size than the one for general subcubic graphs.

6 Future Work

One possible direction of research is that one tries to overcome the bound on the number of reusable vertices of $2.5j$. There are two main possibilities to tackle this problem. Firstly, one could try to modify the greedy P_2-packing by further 'local optimization rules' similar to **Rule 1** and **Rule 2**. Secondly, due to newly invented reduction rules it might be possible that our lower bound examples do not occur anymore. This could lead to an improvement of Theorem 4.
It would be nice to derive smaller kernels than $7k$ or $1.5k$ for P_2-PACKING or TOTAL EDGE COVER, resp., in view of the mentioned lower bound results [3].

A closely related problem is MAXIMUM P_3-PACKING for which R. Hassin and S. Rubinstein [15] found a $\frac{3}{4}$-approximation. We are trying to apply extremal combinatorial methods to save colors for P_d-packings for $d \geq 3$. First results seem to be promising. So, a detailed combinatorial (extremal structure) study of (say graph) structure under the perspective of a specific combinatorial problem seems to pay off not only for kernelization (see [9]), but also for iterative approaches. In view of new kernelization results for quite general packing problems [26], it would be also interesting to apply these ideas in broader contexts.

Developing exact algorithms for MAXIMUM P_2-PACKING would be interesting. Dynamic programming yields an $\mathcal{O}^*(2^n)$-algorithm. Enumerating all possible midpoints takes $\sum_{i=1}^{n/3} \binom{n}{i} \subseteq \mathcal{O}^*(1.8899^n)$. By Theorem 2, TOTAL EDGE COVER can be solved in time $\mathcal{O}^*(1.8899^{1.5k}) \subseteq \mathcal{O}^*(2.5981^k)$. Improving on exact, non-parameterized algorithmics would also improve on the parameterized algorithm for TOTAL EDGE COVER. Alternatively, finding for example a search-tree algorithm for TOTAL EDGE COVER would be interesting.

We finally mention that H. Fernau, J. Kneis and P. Rossmanith could show that also the general TEST COVER problem is in $\mathcal{FPT}$, a bit surprising in view of the fact that the quite similar FEATURE SET problem is W[2]-complete [5, 6]. However, the general algorithm is far from practical and needs to be improved.

References

1. C. Bazgan, R. Hassin, and J. Monnot. Approximation algorithms for some vehicle routing problems. *Discrete Applied Mathematics*, 146:27–42, 2005.
2. K. M. J. De Bontridder, B. V. Halldórsson, M. M. Halldórsson, J. K. Lenstra, R. Ravi, and L. Stougie. Approximation algorithms for the test cover problem. *Math. Progr., Ser. B*, 98:477–491, 2003.
3. J. Chen, H. Fernau, Y. A. Kanj, and G. Xia. Parametric duality and kernelization: lower bounds and upper bounds on kernel size. *SIAM Journal on Computing*, 37:1077–1108, 2007.
4. J. Chen, H. Fernau, D. Ning, D. Raible, and J. Wang. A parameterized perspective on P_2-packings. Technical Report 0804.0570, ArXiv, `http://arxiv.org/abs/0804.0570`, 2008.
5. C. Cotta and P. Moscato. On the parameterized complextiy of problems related with feature identification for gene expression data mining techniques. *Bioinformatics*, 1:1–8, 2002.

6. C. Cotta and P. Moscato. The k-feature set problem is W[2]-complete. *Journal of Computer and System Sciences*, 67:686–690, 2003.
7. F. Dehne, M. Fellows, H. Fernau, E. Prieto, and F. Rosamond. NONBLOCKER: parameterized algorithmics for MINIMUM DOMINATING SET. In J. Štuller, J. Wiedermann, G. Tel, J. Pokorný, and M. Bielikova, editors, *Software Seminar SOFSEM*, volume 3831 of *LNCS*, pages 237–245. Springer, 2006.
8. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.
9. V. Estivill-Castro, M. R. Fellows, M. A. Langston, and F. A. Rosamond. FPT is P-time extremal structure I. In H. Broersma, M. Johnson, and S. Szeider, editors, *Algorithms and Complexity in Durham ACiD 2005*, volume 4 of *Texts in Algorithmics*, pages 1–41. King's College Publications, 2005.
10. M. Fellows. Parameterized complexity: the main ideas and connections to practical computing. *Electronic Notes in Theoretical Computer Science*, 61, 2002.
11. H. Fernau. *Parameterized Algorithmics: A Graph-Theoretic Approach*. Habilitationsschrift, Universität Tübingen, Germany, 2005.
12. H. Fernau and D. F. Manlove. Vertex and edge covers with clustering properties: Complexity and algorithms. In *Algorithms and Complexity in Durham ACiD 2006*, pages 69–84. King's College, London, 2006.
13. H. Fernau and D. Raible. A parameterized perspective on packing paths of length two. In B. Yang, D.-Z. Du, and C. An Wang, editors, *Combinatorial Optimization and Applications COCOA*, volume 5165 of *LNCS*, pages 54–63. Springer, 2008.
14. T. Gallai. Über extreme Punkt-und Kantenmengen. *Ann. Univ. Sci. Budapest, Eötvös Sect. Math.*, 2:133–138, 1959.
15. R. Hassin and S. Rubinstein. An approximation algorithm for maximum packing of 3-edge paths. *Information Processing Letters*, 63:63–67, 1997.
16. R. Hassin and S. Rubinstein. An approximation algorithm for maximum triangle packing. *Discrete Applied Mathematics*, 154:971–979; 2620 [Erratum], 2006.
17. P. Hell and D. G. Kirkpatrick. Star factors and star packings. Technical Report 82-6, Computing Science, Simon Fraser University, Burnaby, B.C. V5A1S6, Canada, 1982.
18. J. E. Hopcroft and R. M. Karp. An $n^{5/2}$-algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.
19. S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoretical Computer Science*, 289:997–1008, 2002.
20. D. G. Kirkpatrick and P. Hell. On the completeness of a generalized matching problem. In *ACM Symposium on Theory of Computing STOC*, pages 240–245, 1978.
21. A. Kosowski, M. Małafiejski, and P. Żyliński. Packing three-vertex paths in a subcubic graph. In Stefan Felsner, editor, *2005 European Conference on Combinatorics, Graph Theory and Applications (EuroComb '05)*, volume AE of *DMTCS Proceedings*, pages 213–218. Discrete Mathematics and Theoretical Computer Science, 2005.
22. I. Koutis. Faster algebraic algorithms for path and packing problems. In L Aceto, I Damgård, L Ann Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, volume 5125 of *Lecture Notes in Computer Science*, pages 575–586. Springer, 2008.
23. Y. Liu, S. Lu, J. Chen, and S.-H. Sze. Greedy localization and color-coding: improved matching and packing algorithms. In H. L. Bodlaender and M. Langston,

editors, *International Workshop on Parameterized and Exact Computation IWPEC*, volume 4169 of *LNCS*, pages 84–95. Springer, 2006.
24. S. Micali and V. V. Vazirani. An $\mathcal{O}(\sqrt{|V|}|E|)$ algorithm for finding maximum matchings in general graphs. *Symposium on Foundations of Computer Science*, pages 17–27. IEEE Press, 1980.
25. J. Monnot and S. Toulouse. The P_k partitioning problem and related problems in bipartite graphs. in J. van Leeuwen *et al.*, editors, *Software Seminar SOFSEM 2008*, volume 4362 of *Lecture Notes in Computer Science*, pages 422–433. Springer, 2007.
26. H. Moser. A problem kernelization for graph packing. In M. Nielsen et al., editors, *Software Seminar SOFSEM*, volume 5404 of *LNCS*, pages 401–412. Springer, 2009.
27. J. Plesník. Equivalence between the minimum covering problem and the maximum matching problem. *Discrete Mathematics*, 49:315–317, 1984.
28. E. Prieto and C. Sloper. Either/or: Using vertex cover structure in designing FPT-algorithms—the case of k-internal spanning tree. In *Proceedings of WADS 2003, Workshop on Algorithms and Data Structures*, volume 2748 of *LNCS*, pages 465–483. Springer, 2003.
29. E. Prieto and C. Sloper. Looking at the stars. In R. Downey, M. Fellows, and F. Dehne, editors, *International Workshop on Parameterized and Exact Computation IWPEC 2004*, volume 3162 of *LNCS*, pages 138–148. Springer, 2004.
30. J. Wang and Q. Feng. An $O^*(3.52^{3k})$ parameterized algorithm for 3-set packing. In M. Agrawal et al., editors, *Theory and Applications of Models of Computation TAMC*, volume 4978 of *LNCS*, pages 82–93. Springer, 2008.
31. J. Wang, D. Ning, Q. Feng, and J. Chen. An improved parameterized algorithm for a generalized matching problem. In M. Agrawal et al., editors, *Theory and Applications of Models of Computation TAMC*, volume 4978 of *LNCS*, pages 212–222. Springer, 2008.