Between Packable and Arbitrarily Packable Graphs: Packer–Spoiler Games

Wayne Goddard*
School of Geological and Computer Sciences
University of Natal, Durban
South Africa

Grzegorz Kubicki†
Department of Mathematics
University of Louisville, Louisville
U.S.A.

ABSTRACT. In a packer-spoiler game on a graph, two players jointly construct a maximal partial F-packing of the graph according to some rules, where F is some given graph. The packer wins if all the edges are used up and the spoiler wins otherwise. The question of which graphs are wins for which player generalizes the questions of which graphs are F-packable and which are randomly F-packable. While in general such games are NP-hard to solve, we provide partial results for $F = P_3$ and solutions for $F = 2K_2$.

1 Introduction

A *packing* of a graph G with graph F is a decomposition of the edge set of G into copies isomorphic to F. If such a packing exists we say that G is *F-packable*. We are interested in the case where F is fixed; this concept is much studied.

Ruiz [8] introduced the concept of randomly packable graphs. A *partial F-packing* of G is a packing of a subgraph of G. Then the graph G is *randomly F-packable* if any partial F-packing of G can be extended to a (full) packing of G. We prefer the term *arbitrarily packable.* Thus, a

[0]Research supported by Foundation for Research Development
[0]Research supported by Office of the Vice President for Research and Development, University of Louisville

packable graph is one where if one is careful a packing can be found. An arbitrarily packable graph is one where one can be carefree when packing.

For the two simplest choices for the graph F, the characterizations are as follows. Here $\Delta(G)$ denotes the maximum degree of G and $q(G)$ the size (number of edges) of G:

Theorem 1 *[2, 4] A graph G is*
(a) P_3-packable iff every component has even size.
(b) $2K_2$-packable iff $\Delta(G) \leq q(G)/2$ and $G \neq K_2 \cup K_3$.

Theorem 2 *[8] A graph G is*
(a) arbitrarily P_3-packable iff every component is the 4-cycle C_4 or an even star $K(1, 2s)$.
(b) arbitrarily $2K_2$-packable iff G is an even number of disjoint edges $2sK_2$, or the disjoint union of two isomorphic stars $2K(1, s)$, or some small graph (C_4, K_4, $2K_3$ or $K_3 \cup K(1, 3)$).

These theorems show that it is very easy to find a packable graph, but very hard to find one that is arbitrarily packable. Further work on arbitrarily packable graphs can be found in [1, 3, 5] inter alia.

In the case of $F = P_3$ and $2K_2$ there is also a link between packing and matching. Since a P_3 in G corresponds to an edge in the line graph $L(G)$ of G, a graph has a P_3-packing if and only if its line graph has a perfect matching. A graph has a $2K_2$-packing if and only if the complement of the line graph has a perfect matching. A characterization of which graphs are arbitrarily matchable (that is, every partial matching is extendable to a perfect matching), is due to Sumner [9]. This says that each component must be either a complete graph or a balanced complete bipartite graph.

Now, the concepts of packable and arbitrarily packable graphs may also be thought of as solitaire games. In one game the player P tries to find a maximal packing that uses all the edges. The games where P wins are the packable ones. In another game, one player S tries to find a maximal packing that does *not* use all the edges. The graphs where S loses are the arbitrarily packable ones.

In this paper we introduce a generalization of these games. We consider two players called P and S. They are supplied graphs F and G. According to some rules, discussed below, a maximal partial F-packing of G is constructed. The player P wins if and only if all the edges of G are used up, i.e., if and only if the partial packing is a packing. We think of P as the packer and S as the spoiler.

The natural game is the one where P and S alternate. This game is hard to analyse, even for F the path on three vertices. In fact, that game is a special case of the "alternating maximum weighted matching" problem (Problem [GP8] in the book [7]), which is PSPACE-complete, but it is unclear whether this special case remains PSPACE-complete.

Rather, we are interested in games which measure how close to not packable or to arbitrarily packable the graph is. For this purpose we consider three types of games: hand-over games, intervention games, and completion games. These are described in Section 2. Then in Section 3 we discuss these games for $F = P_3$ and in Section 4 we discuss them for $F = 2K_2$. This discussion makes extensive use of Theorems 1 and 2. In Section 5 we discuss the complexity of these games for general F, noting that in general finding optimal strategies is NP-complete.

2 Six Packer–Spoiler Games

Hand-over games. In these, one player makes all his moves and then hands over to the other player.

> *Game 1:* P chooses and removes as many copies as desired, paying $1 each time, and then hands over to S.

> *Game 2:* S chooses and removes as many copies as desired, paying $1 each time, and then hands over to P.

Obviously, in Game 1 the player P makes enough moves to leave an arbitrarily packable graph, and in Game 2 the player S makes enough moves to leave an unpackable graph. How much P or S spends is a measure of how close to arbitrarily packable or to non-packable the graph is.

Note that the solution to Game 1 is not simply to leave the largest subgraph that is arbitrarily packable, but rather the largest arbitrarily packable subgraph which is obtainable by removing copies of F. For example, on the graph $G = K_4$ with $F = P_3$, the largest arbitrarily packable subgraph is C_4, but P cannot leave this subgraph.

Intervention games. In these, one player has the power to intervene whenever she desires.

> *Game 3:* At each stage P has the option of choosing the next copy. Exercising that option costs $1. Otherwise S chooses the next copy.

Game 4: At each stage S has the option of choosing the next copy. Exercising that option costs \$1. Otherwise P chooses the next copy.

It should be noted that for Games 1 and 3, P wins by paying \$0 if and only if the graph is arbitrarily packable; and P loses no matter what the payment is if and only if the graph is not packable. The amount of money that P pays in Game 3 is at most that paid in Game 1. Analogous statements hold for Games 2 and 4.

Completion games. In these, the players cooperate to choose a copy.

Game 5: P chooses an edge in the graph and S chooses a copy of F containing the chosen edge.

Game 6: S chooses an edge in the graph and P chooses a copy of F containing the chosen edge.

However, Game 6 is simply a win for P if and only if the original graph was packable.

3 The Path with Two Edges

Despite the simplicity of $F = P_3$, several of the games seem hard to analyse.

3.1 Games 1 & 3: P hands over or intervenes

In Game 1 P must leave an arbitrarily P_3-packable graph. The densest arbitrarily packable graph is the union of 4-cycles. So, if such a spanning subgraph exists, P must leave it if possible. This is the case for dense graphs such as the complete graph K_n for $n \geq 5$, the cubes Q_n for $n \geq 4$ and the balanced complete bipartite graphs. We omit the details. At the other extreme, there is a linear-time algorithm for P for trees.

Observation 3 *For Game 1 and $F = P_3$, there is a linear-time algorithm for P in trees.*

Proof: A *leaf* is an edge incident to an end-vertex. A *penultimate vertex* is one which is not an end-vertex but at most one of its neighbors is not an end-vertex. If the penultimate vertex v has a neighbor that is not an end-vertex, then that neighbor is the *interior* neighbor of v and the edge joining them is the *interior* edge of v.

Let T be a tree with even size. We orient each edge $e = uv$ from u to v such that the component of $T - e$ containing v has odd size. Then every vertex in the orientation has even in-degree. Also, in any P_3-packing of T, the edge uv lies in a P_3 centered at v.

A subgraph is arbitrarily packable if and only if it is the union of even stars. So the packer must leave a subgraph of T where the vertices are of two types: "even sinks" and "tails". An even sink has even indegree and zero outdegree; a tail has zero indegree and outdegree 1. Furthermore, if H is any such subgraph, then $T - E(H)$ is P_3 packable: every vertex has even indegree in T and in H and hence in $T - E(H)$, and the size of a component is the sum of the indegrees.

Thus, we need to determine the subgraph of T with maximum size where every vertex is an even sink or a tail.

This can be done with a typical depth-first-search tree algorithm. For a tree T_v rooted at v define three functions:

- $f(T_v)$ is the maximum size of a subgraph in which every vertex is an even sink or a tail;
- $g(T_v)$ is the maximum size of a subgraph in which every vertex is an even sink or a tail and moreover v is isolated;
- $h(T_v)$ is the maximum size of a subgraph in which every vertex except the root is an even sink or a tail, and the root is an odd sink.

We want to know the value $f(T)$.

Assume we know the values of f, g and h for the subtrees rooted at the children of v and want to know the value for the tree rooted at v. Then to calculate $f(T_v)$ we consider the two options for v. Suppose v is to be an even sink. Then, if the arc from a child w is present w must be isolated in T_w, and if the arc is absent w must be an even sink or tail in T_w. So, if $g(T_w) = f(T_w)$ we want the arc, if $g(T_w) \leq f(T_w) - 2$ we do not want the arc, and if $g(T_w) = f(T_w) - 1$ it is immaterial. It is straightforward to determine the optimal number of arcs subject to the restriction that there is an even number of them. Suppose v is to be a tail; say adjacent to w. Then w must be an odd sink in its subtree, and every other child is free. So we find the child w with maximum value of $h(T_w) - f(T_w)$.

To calculate $g(T_v)$ one simply adds the values of $f(T_w)$ for the children. The calculation of $h(T_v)$ is similar to that of $f(T_v)$. □

In Game 3 the packer has the power to intervene. Since the packer must make the next move if there exists a nonleaf bridge, it follows that, for trees, P must make enough moves to leave an arbitrarily packable graph. Hence the strategy for trees for P is the same as in Game 1, and the cost the same.

For general graphs we have only a weak bound.

Observation 4 *For Game 3 and $F = P_3$, if P can win, then P can win by paying at most $2n$ dollars (where n is the order of the graph).*

Proof: One strategy for P is to let S choose the next copy unless there exists a copy F of P_3 whose removal would create an odd-size component. When such a copy exists, P chooses the next two P_3's to use up the edges of F and leave a packable graph: this is possible since the graph is packable. This action of P increases the number of components, so P needs to intervene at most n times. □

It is unclear what the constant in front of n is in the worst case.

3.2 Games 2 & 4: S hands over or intervenes

In Game 2 the spoiler must leave a component of odd size. If the graph contains a nonleaf bridge this is easy. Hence a tree that is not a star is easily won by S.

For general graphs, the following result shows that the minimum is related to the (vertex) connectivity κ.

Observation 5 *For Game 2 and $F = P_3$, if G is P_3-packable, then S has to pay at least $\kappa - 1$ dollars.*

Proof: Let Z be a minimum collection of edge-disjoint P_3's such that $G - E(Z)$ is not packable; $|Z| = z$. Let X denote the set of the centers of the P_3's in Z. If $G - X$ is disconnected, then $\kappa \leq z$ and we are done. So suppose not. Since $G - E(Z)$ has at least two nontrivial (odd-size) components, there exists a nontrivial component A of $G - E(Z)$ consisting entirely of vertices of X. In particular, if a denotes the order of A then $2 \leq a \leq z$.

The total degree sum of the vertices in A (as measured in G) is at most the sum of twice the number of edges with both ends in A, added to the number of edges with exactly one end in A. Hence the total degree sum of A is at most $2\binom{a}{2} + 2z$. Thus the average degree of A is at most $a-1+2z/a$. Since $2 \leq a \leq z$, it follows by calculus that the average degree of A is at most $z+1$. Hence $\kappa \leq \delta \leq z+1$. □

Equality holds for example for r-connected r-regular graphs of even size for r odd. (The optimal strategy for S is to isolate one edge.)

Solving Game 2 for P_3 is equivalent to asking about extendable graphs. In fact, the amount S must spend is 1 dollar more than the maximum i

such that the line graph $L(G)$ is i-extendable (i.e., every partial matching of i edges can be extended to a perfect matching). The complexity of determining for an arbitrary graph H the maximum i such that H is i-extendable remains unresolved. Even for our case where H is a line graph, it has neither been shown to be polynomial-time solvable nor to be NP-complete.

In Game 4 the spoiler has the power to intervene.

Observation 6 *For Game 4 and $F = P_3$, if S can win, then S can win by paying at most 2 dollars.*

Proof: The spoiler waits until there is a graph G' in which there exists a copy F of P_3 whose removal would leave an arbitrarily packable graph. S then intervenes. We need to show that S can destroy the packability of G' with at most 2 moves. This is easy if there is a component of $G' - E(F)$ that only one edge of F touches (since then there is a non-leaf bridge). Otherwise a bit more work is required. □

There are graphs where S pays \$1 and must make the first move, and ones where S pays \$1 and must wait. Indeed, the exact characterization of which graphs S pays \$1 for is unclear.

3.3 Game 5: S completes

In this game P picks an edge and S picks an edge which completes the P_3. This game also seems hard; even who wins in the complete graph is unclear. We discuss Game 5 only for trees. We say that a vertex is even or odd depending on its degree.

Observation 7 *For Game 5 and $F = P_3$, in forests, an optimal strategy for P is always to choose any leaf incident with an even penultimate. If no such leaf exists, then S can force a win.*

Proof: We prove this by a sequence of claims. We assume we have a forest where every component has even size (otherwise P has already lost).

1. *To win, P may only choose a leaf.* Suppose P chooses edge uv where the component containing v is the one with odd size. Then S must be forced to take the next edge in that component (otherwise there will be a component with odd size left and P is lost). This means that the component containing u is empty; that is, u is an end-vertex.

2. *To win, P may only choose a leaf whose interior end is even.* Suppose P chooses edge uv where u is an end-vertex. Let x be any other neighbor of v. If S chooses the edge xv, then the component containing v so created must have even size. Hence v has even degree.

3. *If every penultimate vertex is odd than S wins.* We will show that no matter what P does, S can always ensure that there remains a component without an even penultimate. A component cannot disappear without going through the stage of there being only two edges left—at which stage it has an even penultimate. Thus P cannot remove the component entirely.

   Consider P's next move. We know that this must be a leaf uv where v is even. Since v is not a penultimate it has at least two neighbors which are not end-vertices. Since v has degree at least 4, it has another neighbor, call it w. If S selects vw, then the component containing v that remains is without an even penultimate.

4. *If e is a leaf incident with a penultimate of even degree, and P can win, then P can win by playing e next.* This is clearly true for a star, so suppose that $e = uv$ with v the penultimate having x as a non-leaf neighbor.

   Follow P's winning strategy until the first edge incident with v is chosen by one of the players. This cannot be S choosing vx, since then P has lost. So it must be P choosing some edge. If it is P taking vx then (by the results above) x must be an end-vertex at that stage, and so P could now choose e instead. Hence we may assume that the next edge incident with v that is chosen is e and P does the choosing. But then P could have chosen e at first move: it does not interfere with P's options in the winning strategy, and can only reduce S's choices. □

4 Two Disjoint Edges

The graph $2K_2$ seems in general easier to analyze than P_3. Note that a maximal partial packing leaves a star. We denote the cardinality of the largest matching in G by $m(G)$ or simply m.

4.1 Games 1 & 3: P hands over or intervenes

In Game 1 the player P must remove some copies of $2K_2$ and leave either $2K(1,s)$ or $2sK_2$ or some small graph (listed in Theorem 2).

Theorem 8 *For Game 1 and $F = 2K_2$, there is a polynomial-time algorithm to determine the optimal strategy for P.*

Proof: Let G be a graph with $\Delta \leq q/2$. It suffices to show that one can compute the minimum number of moves to leave a matching, the minimum number to leave two equal stars, and the minimum number to leave one of the small graphs described in Theorem 2.

To leave a matching. We claim that the maximum matching one can leave has size the minimum of $2\lfloor m/2 \rfloor$ and $q - 2(\Delta - 1)$. This quantity is certainly an upper bound: any vertex of degree Δ has to have its degree reduced to at most 1 and this takes at least $\Delta - 1$ moves.

But the bound is attainable. If $m \leq 1$ this is trivial; so assume $m \geq 2$. Let M be a matching of cardinality $2\lfloor m/2 \rfloor$ incident to as many vertices of degree Δ as possible. If $G - M$ has a $2K_2$ packing, then we are done by Theorem 1. Otherwise $G - M$ has a vertex v of degree at least $\Delta - 1$. (The case where the edges of $G - M$ form $K_3 \cup K_2$ is easily dealt with, so we assume otherwise.) We may assume that v is incident with an edge of M and that v has degree Δ in G. So $\Delta - 1 > (q - m)/2$ and thus $m > q - 2(\Delta - 1)$; also v is the unique vertex of degree Δ in G. In this case, P removes $\Delta - 1$ copies of $2K_2$, each using an edge incident with v, as follows: the edges outside M are used up first, and then some edges in M are used. When $\Delta - 1$ copies of $2K_2$ have been removed, we are left with a matching.

To leave two stars. It suffices to calculate for each pair $\{x, y\}$ of vertices the size of the largest pair of stars centered at x and y that P can leave. Define T_{xy} as the largest subgraph consisting of two equal-sized disjoint stars centered at x and y. Say x has a private neighbours in G, y has b private neighbours and they have c common neighbours, $a \geq b$. Then a bit of calculation shows that T_{xy} has stars each with size the minimum of $b + c$ and $\lfloor (a + b + c)/2 \rfloor$. If $G - E(T_{xy})$ has a $2K_2$-packing then we are done.

Otherwise, $G - E(T_{xy})$ has a vertex v of large degree. Then to leave a maximum star-pair centered at $\{x, y\}$, P must delete copies of $2K_2$ from $G - E(T_{xy})$ each using an edge incident with v. When there is only a star centered at v left in $G - E(T_{xy})$, P switches to removing copies with one edge incident with v and the other edge alternately with x and with y.

So by considering each pair $\{x, y\}$ of vertices in turn, we can determine the largest star-pair that P can leave.

To leave a small graph. Whether or not a particular small graph can be left by P is easily calculated. For example, one can simply try all possible isomorphic subgraphs.

Thus there is a polynomial-time algorithm for determining P's optimal choices. □

In Game 3 we believe that a near-optimal strategy for P is to intervene only when S is threatening to leave an unpackable graph. While we have partial results to support this, they are messy and we omit them.

4.2 Games 2 & 4: S hands over or intervenes

In Game 2 the player S must remove some copies of $2K_2$ and leave either a graph with high maximum degree or $K_3 \cup K_2$.

Theorem 9 *For Game 2 and $F = 2K_2$, for $\Delta \geq 3$, if S can win then S can win by paying $q/2 - \Delta + 1$ dollars, and this is best possible.*

Proof: *Upper bound.* To make $q/2 - \Delta + 1$ moves, S considers any vertex v of maximum degree Δ. S then removes edges noincident with v such that the resultant graph has q' edges, with $q' < 2\Delta$, but the maximum degree is still Δ. This is achieved as follows.

The spoiler removes $2K_2$ from $G - v$ arbitrarily until there are $\Delta + 2$ edges not incident with v. Now S must remove a copy to avoid leaving the star on Δ edges. This is possible, since for $\Delta \geq 4$ there is at most one vertex with degree Δ outside v and S takes a copy of $2K_2$ incident with this vertex (and the case $\Delta = 3$ is easily argued). Since we have Δ edges not incident with v and these do not induce a star, there is a copy of $2K_2$ that can be removed. Hence S can leave overall a graph with $q' = 2\Delta - 2$ edges and a vertex of degree Δ.

Lower bound. S must leave either a graph with too high maximum degree or the graph $K_3 \cup K_2$. To leave the former requires $q' < 2\Delta'$. Hence if S makes r moves, we must have that $q - 2r = q' < 2\Delta' \leq 2\Delta$, so that $r > q/2 - \Delta$. Hence S must make at least that number of moves. To leave $K_3 \cup K_2$ requires $q/2 - 2$ moves. □

The $\Delta = 2$ case. It is easily checked that if S can win, S can leave $K_3 \cup K_2$ if G contains a triangle, and otherwise must leave P_3.

In Game 4 the player S must intervene so that the resultant graph is not packable. We show that Game 4 costs the same as Game 2 if the graph is sparse, and less if the graph is dense.

For a graph G with even size let

$$g(G) = \min\{\lfloor m/2 \rfloor, q/2 - \Delta + 1\}$$

The importance of this function lies in the fact that: if G has no isolates and is not $K_3 \cup K_2$ then G is $2K_2$-packable if and only if $g(G) > 0$. Also, if $G' = G - E(2K_2)$ then $g(G')$ is $g(G)$ or $g(G) - 1$.

Theorem 10 *For Game 4 and $F = 2K_2$, the minimum number of moves for S is at least $g(G) - 1$ and at most* $\max\{g(G), 2\}$.

Proof: We show first that S can spend at most $g(G)$ dollars by waiting for a graph G' that contains a copy F of $2K_2$ whose removal would leave an arbitrarily packable graph. There are three cases:

(a) *$G' - E(F)$ is a double star.* Then it can be shown that one can remove two copies of $2K_2$ without reducing the maximum degree of the graph. So by spending two dollars G' can be converted into an unpackable graph.

(b) *$G' - E(F)$ is one of the four small graphs.* Then by considering each possibility for G' in turn it can be shown that that by spending two dollars G' can be converted into an unpackable graph.

(c) *$G' - E(F)$ is a matching on m' edges.* Then G' has $m' + 2$ edges and so, since it is not arbitrarily packable, by removing at most $m'/2$ pairs of edges one can leave an unpackable graph. Thus, by spending at most $m'/2$ dollars S' can leave an unpackable graph.

Clearly $m'/2 \leq m/2$ since $m' \leq m$. Furthermore, $m' = q(G' - E(F)) \leq q(G) - 2(\Delta(G) - 1)$, since the vertex of maximum degree in G has to be reduced at least $\Delta(G) - 1$ times to get to $G' - E(F)$. It follows that $m'/2 \leq g(G)$, and the upper bound is established.

We next show that S must spend at least $g(G) - 1$ dollars. At some stage an unpackable graph is reached: in this the value of g is at most 1. If S intervenes, she can reduce g by at most 1 for one dollar. It therefore suffices to show that the packer can always choose a copy of $2K_2$ in a graph H such that $g(H - E(F)) = g(H)$.

There are three cases:

(a) $m(H) < q(H) - 2\Delta(H) + 2$. Then we need to find a $2K_2$ whose removal does not decrease the matching number. Since $q - m > 2\Delta - 2 \geq \Delta$, the edges not in a maximum matching cannot all be adjacent. So P can choose two edges not in a maximum matching.

(b) $m(H) > q(H) - 2\Delta(H) + 2$. Then we need to find a $2K_2$ whose removal reduces the maximum degree and hence does not decrease the difference $q - 2\Delta$. Since $q < m + 2(\Delta - 1)$, there cannot be two vertices of degree Δ. So P can choose any copy of $2K_2$ containing an edge incident to a vertex of maximum deree.

(c) $m(H) = q(H) - 2\Delta(H) + 2$. Then we need to find a $2K_2$ whose removal decreases neither the matching number m nor the difference $q - 2\Delta$. If there is a unique vertex v of maximum degree, then take any edge which is neither adjacent to v nor in a specific maximum matching, and pair with an edge incident with v.

Otherwise, suppose there are two vertices of maximum degree. Then the edges of H can be decomposed into $2K(1, \Delta - 1)$ and a maximum matching. Hence P can remove two edges incident with the large vertices without decreasing the matching number, and the value of g is unchanged. □

4.3 Game 5: S completes

This game has a simple solution.

Theorem 11 *For Game 5 and $F = 2K_2$, P wins iff the graph is $2K_2$-packable.*

Proof: Claim: P wins by playing the following strategy, except when $q = 6$ and three of the six edges form a triangle:

> *Choose any edge e incident with as many vertices of degree $q/2$ as possible.*

(That is, if there is no vertex of degree $q/2$ then P can do what he likes.)

Proof of claim by induction. Assume we are not in the case where $q = 6$ and there is a triangle. Suppose P chooses the edge e by the above rule and then S chooses edge f. If the resultant graph G' (after removal of the two disjoint edges e and f) is $2K_2$-packable, then P can win by inductive hypothesis. So suppose the resultant graph G' is not $2K_2$-packable.

Then there must exist a vertex x in G' which has degree more than $q(G')/2$. This means that x has degree $q/2$ in G and is not incident with e or f. If only one end of e, say y, has degree $q/2$, then x and y are nonadjacent (by the choice of e) and together the two vertices x and y are incident with all q edges, which means that f must be incident with one of them, a contradiction. If both ends of e have degree $q/2$, then since x is not incident with f it can have degree at most 2. This means that $q \leq 4$ and we can check by hand that P's strategy works.

If $q = 6$ and there is a triangle, then one rule that works is to take an edge in the triangle that is incident to as many vertices of degree 3 as possible. It is easily verified that, after S's move, what remains in every case is $2K_2$-packable. □

5 The General Game

A deep result is the following:

Theorem 12 *[6] For a fixed graph F containing a component of at least 3 edges, then determining whether graph G is F-packable is NP-hard.*

Another result is the following:

Theorem 13 *[3] For a fixed graph F, there is a polynomial-time algorithm for determining whether graph G is arbitrarily F-packable.*

These results show that several of the games are hard. For example, Game 1 is a win for P if and only if the graph is packable, and Game 2 is a win for S for 0 dollars if and only if the graph is not packable. Similarly, Game 3 is a win for P if and only if the graph is packable. Hence these games are intractable.

Game 4 (spoiler intervention) is not always intractable. For example, consider the game with K_3-packing and spoiler intervention. Here the spoiler must intervene at most once and we can give the optimal strategy for S. The optimal strategy for P, on the other hand, involves finding a K_3-packing, which is NP-hard.

The strategy for the spoiler is as follows. As P removes copies, the spoiler repeatedly checks whether there exists a triangle whose removal leaves an arbitrarily packable graph. By theorem of Beineke et al. [1], a graph is arbitrarily K_3-packable if and only if every edge is in exactly one triangle.

Suppose that xyz is a triangle T such that the removal of the edges of T leaves an arbitrarily packable graph, but with the edges of T the graph is not arbitrarily packable. Then by the characterization, there is an edge of T that is in another triangle. Say this triangle is xym. Consider removing the edges of T. Then by the characterization the edge xm is in a unique triangle; say xmt.

Now the optimal strategy is for S to remove the edges of the triangle xym. The triangle containing the edge mt is destroyed, but the edge remains and the resultant graph is not packable. So S need only intervene once. Whether P can force this intervention depends on whether the graph is K_3-packable or not.

References

[1] L.W. Beineke, W. Goddard and P. Hamburger, Random packings of graphs, Discrete Math. 125 (1994), 45–54.

[2] Y. Caro, Chromatic decomposition of a graph, Ars Combin. 21 (1986), 29–32.

[3] Y. Caro, J. Rojas and S. Ruiz, A forbidden subgraphs characterization and a polynomial algorithm for randomly decomposable graphs. Czech. Math. J. 46 (1996), 413–419.

[4] Y. Caro and J. Schönheim, Decompositions of trees into isomorphic subtrees, Ars Combin. 9 (1980), 119–130.

[5] Y. Caro, A Sebo and M. Tarsi, Recognising greedy structures, J. Algorithms 20 (1996) 137–156.

[6] D. Dor and M. Tarsi, Graph decomposition is NPC — a complete proof of Holyer's conjecture. SIAM J. Comput. 26 (1997) 1166–1187.

[7] M.S. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W.H. Freeman, New York, 1982.

[8] S. Ruiz, Randomly decomposable graphs, Discrete Math. 57 (1985), 123–128.

[9] D.P. Sumner, Randomly matchable graphs, J. Graph Theory 3 (1979), 183–186.