The Generalized Matcher Game

Anna Bachstein, Wayne Goddard, Connor Lehmacher

School of Mathematical and Statistical Sciences, Clemson University
Department of Mathematics, Vanderbilt University

Abstract

Recently the matcher game was introduced. In this game, two players create a maximal matching by one player repeatedly choosing a vertex and the other player choosing a \( K_2 \) containing that vertex. One player tries to minimize the result and the other to maximize the result. In this paper we propose a generalization of this game where \( K_2 \) is replaced by a general graph \( F \). We focus here on the case of \( F = P_3 \). We provide some general results and lower bounds for the game, investigate the graphs where the game ends with all vertices taken, and calculate the value for some specific families of graphs.

1 Introduction

Fix some graph \( F \) with a designated “root” vertex \( r \). Given a graph \( G \), two players cooperate in choosing a collection of vertex-disjoint subgraphs of \( G \) each isomorphic to \( F \). Each round, one player \textit{initiates} by choosing a vertex \( v \), subject to the constraint that the unused portion of \( G \) contains at least one (not necessarily induced) copy of \( F \) with vertex \( v \) corresponding to vertex \( r \). The other player \textit{responds} by choosing one such copy of \( F \) within the unused portion of \( G \). Vertices can only be used once. This process continues until the unused portion of \( G \) does not contain a copy of \( F \); that is, they choose a maximal collection.

One player tries to maximize the number of copies of \( F \) taken. The other player tries to minimize this number. We call these players \textit{Maximizer} and \textit{Minimizer}. Thus there are two versions, depending on who initiates and who responds. We define the \textit{value} of the game as the number of copies of \( F \) taken with optimal play by both players.

This game is a generalization of the \textit{matcher game} introduced in [4]. That game is where the subgraph \( F \) is \( K_2 \). For example, it was shown in [4] that
if Maximizer is the responder, then the value of the game is just the matching number of the graph.

The generalized matcher game has connections to the family of competition parameters or competitive optimization games on graphs. In these, two players alternate to form some structure, with opposing goals. In most of these games the palette of moves is the same for the two players; these situations include the game chromatic number of Gardner [3], the competition independence number of Philips and Slater [7], the domination game of Brešar et al. [1], and the game matching number of Cranston et al. [2]. However, there are some games with an initiation/response model similar to the generalized matcher game, such as the Paint–Correct game of Schauz [8].

In this paper, we investigate the generalized matcher game for the graph $P_3$. Rooted at the center vertex we call it the 2-star or simply the star; rooted at an end-vertex we call it the stripe. Thus we talk of the star-game and the stripe-game.

We proceed as follows. In Section 2 we provide some examples and elementary results. In Section 3 we determine lower bounds for the game in general graphs and in Section 4 we consider the graphs where the game ends with all vertices taken. In Section 5 we consider some specific families of graphs including grids. Finally in Section 5 we consider the alternative game where there is no “root” vertex.

2 Examples

For a graph $G$, we define a $P_3$-packing as a collection of vertex-disjoint copies of $P_3$ in $G$. Further, we denote by $\mu(G)$ the maximum size of a $P_3$-packing of $G$, and say that a graph $G$ of order $n$ is $P_3$-packable if $\mu(G) = n/3$. This concept generalizes matching and is well-studied. For example, Kaneko et al. [5] showed that if $G$ is a 2-connected claw-free graph of order $n$ a multiple of 3, then $G$ is $P_3$-packable. Earlier, Kirkpatrick and Hell [6] showed that the parameter $\mu(G)$ is NP-complete to compute.

Because the star-and stripe-games produce maximal $P_3$-packings, the parameter $\mu(G)$ provides an immediate upper bound on the value of both games. At
the other extreme is the minimum size of a maximal $P_3$-packing. It is easy to see that that quantity is at least $\mu(G)/3$.

As a first example, consider the complete bipartite graph. Note that up to symmetry, the response is forced. So Maximizer as initiator can ensure a maximum $P_3$-packing and Minimizer as initiator can ensure a minimum maximal $P_3$-packing, regardless of whether it is the star- or stripe-game. Thus:

**Proposition 1** Consider the complete bipartite graph $K_{r,s}$ with $r \leq s$ and $s \geq 2$. The value of the game with Maximizer initiating is $\mu = \min(r, \lceil (s + r)/3 \rceil)$. The value of the game with Minimizer initiating is $\lceil r/2 \rceil$.

Consider next the game played on a path. Assume Maximizer initiates. They can ensure (almost) all the vertices are chosen as follows. For the stripe-game they start by choosing an end-vertex and for the star-game they start by choosing a neighbor of an end-vertex. Minimizer’s response is forced and what remains is a path, whereupon Maximizer repeats the strategy.

So consider the version where Maximizer responds. For the star-game, there is a unique $P_3$ for a given central vertex. Thus the game where Minimizer initiates is equivalent to minimum maximal $P_3$-packing. The arrangement is to skip two vertices, take a $P_3$, skip two vertices, etc. Thus the value of the star-game played on $P_n$ is $\lfloor (n + 2)/5 \rfloor$ if Minimizer initiates.

**Proposition 2** Consider the stripe-game with Maximizer responding. The value of the game played on the path $P_n$ is $\lfloor (n + 1)/4 \rfloor$.

**Proof.** Assume the vertices are numbered from 1 up to $n$. Minimizer can ensure at most the claimed value by playing in succession vertex number 2, 6, 10, etc. If $n \equiv 3$ mod 4, then a final initiation of vertex $n - 1$ is invalid, but (both) $n$ and $n - 2$ are valid final initiations.

To show that Maximizer as responder can ensure at least the claimed value, let $f(j)$ denote the value of the game played on the path $P_j$. If Maximizer chooses vertex $k$, then in most cases Minimizer has two options which in most cases split
the path into two paths. Thus it suffices to show by induction that the recurrence relation

\[
f(n) = 1 + \min \left\{ \begin{array}{l}
f(n-3) \\
f(n-4) \\
\min_{3 \leq k \leq n-2} \max \{ f(k-1) + f(n-k-2), f(k-3) + f(n-k) \}
\end{array} \right\},
\]

with \( f(j) = 0 \) for \( j \leq 2 \) has solution \( g(n) = \lfloor (n+1)/4 \rfloor \). For \( k' = k + 4 \) it holds that \( g(k' - 1) + g(n - k' - 2) = g(k - 1) + g(n - k - 2) \) and that \( g(k' - 3) + g(n - k') = g(k - 1) + g(n - k) \). So it suffices to check the recurrence for say \( 3 \leq k \leq 6 \). From this it follows that if \( g(n) \) satisfies the recurrence \( n \) then so does \( g(n+4) \). So it suffices to check the recurrence for four consecutive values of \( n \), e.g. \( 4 \leq n \leq 7 \). This can be performed by hand or computer. QED

3 General Lower Bounds

3.1 Lower bounds for Maximizer responding

Proposition 3 Consider the star-game with Maximizer responding. If \( G \) is a graph with \( \mu(G) = m \), then the value of the game is at least \( \lceil m/2 \rceil \) and this is sharp.

Proof. For bound: consider a maximum \( P_3 \)-packing \( \mathcal{P} \) of \( G \). We say a copy of \( P_3 \) from \( \mathcal{P} \) is clean if none of its vertices has been used. We claim that Maximizer as responder can ensure that each move reduces the number of clean copies of \( P_3 \) by at most 2. This is immediate except when the initiation vertex \( v \) is chosen in a clean copy of \( P_3 \). Because none of the vertices of that copy of \( P_3 \) has been used, there is an edge to a neighbor of \( v \) within that copy, and Maximizer can use that neighbor. Thus Maximizer can ensure that after \( k \) moves there are still at least \( m - 2k \) clean copies of \( P_3 \). Since the game continues while there is as least one clean copy of \( P_3 \), this means that the game lasts at least \( \lceil m/2 \rceil \) moves.

For optimality: We define the comb-graph \( D_m \) by taking \( m \) copies of \( P_3 \), picking one end-vertex from each copy, and making all the chosen vertices into a path \( K \). See Figure 1 where the vertices of \( K \) are drawn in white. The strategy for Minimizer on the comb-graph \( D_k \) for \( k > 1 \) is to initiate at a vertex in \( K \).
that has one neighbor in $K$. Then Maximizer is forced to respond with a star using the neighbor vertex of $K$ and one degree-2 vertex, leaving as components the comb-graph $D_{k-2}$ and $P_2$. QED

Figure 1: A $P_3$-packable graph $D_6$ with smallest value of game for Maximizer responding

**Proposition 4** Consider the stripe-game with Maximizer responding. If $G$ is a graph with $\mu(G) = m$, then the value of the game is at least $\lceil m/2 \rceil$ and this is sharp.

**Proof.** For bound: consider a maximum $P_3$-packing $\mathcal{P}$ of $G$. We claim that Maximizer as responder can ensure that each move overlaps at most two clean copies of $P_3$ in $\mathcal{P}$. For, suppose the initiation is the end-vertex $a$ of a stripe $abc$ that intersects three clean copies $Q_a, Q_b, Q_c$ of $P_3$ in $\mathcal{P}$. Then vertex $b$ has an unused neighbor $d$ within $Q_b$, so Maximizer can choose the stripe $abd$ instead. Again the game lasts at least $\lceil m/2 \rceil$ moves.

For optimality: consider the comb-graph $D_m$ as in Figure 1 above. The strategy for Minimizer on the comb-graph $D_k$ for $k > 1$ is to initiate on a vertex adjacent to both a leaf and a vertex of degree 2. Maximizer is forced to respond with a stripe using two white vertices, leaving as components $D_{k-2}$ and $P_2$. QED

### 3.2 Lower bounds for Minimizer responding

**Proposition 5** Consider the star-game with Minimizer responding. If $G$ is a graph with $\mu(G) = m$, then the value of the game is at least $\lceil m/3 \rceil$ and this is sharp.

**Proof.** For bound: consider a $P_3$-packing of $G$ with $m$ copies of $P_3$. No matter what the players do, each move can overlap at most three of these copies. So the
game lasts at least $m/3$ moves. In other words, the lower bound follows from the lower bound on the size of a maximal $P_3$-packing noted earlier.

For optimality: consider the “double corona” of the complete graph $K_m$, with $m$ a multiple of 3. That is, take $m$ copies of $P_3$ and add an edge between every two center vertices to form clique $K$. See Figure 2. The initiator has to choose a vertex from $K$. Minimizer can respond by taking two more vertices from $K$. The game stops after $\lceil m/3 \rceil$ moves. QED

![Complete graph]

Figure 2: A $P_3$-packable graph with smallest value of star-game for Minimizer responding

The above result can be improved for a tree:

**Proposition 6** Consider the star-game with Minimizer responding. If $T$ is a tree with $\mu(T) = m$, then the value of the game is at least $\lceil m/2 \rceil$ and this is sharp.

**Proof.** For bound: consider an optimal $P_3$-packing $\mathcal{P}$ of $T$. Consider a vertex $v$ that has at most one non-leaf neighbor. Maximizer initiates at vertex $v$. Then $v$ has at most one edge to a non-leaf, and so Minimizer can overlap at most two stars in $\mathcal{P}$ when responding. Thus the game lasts at least $m/2$ moves.

For optimality: Consider the caterpillar $E_m$ formed from $m$ 2-stars by adding edges so that their centers form a path, called a *spine*. See Figure 3. Maximizer can only initiate at vertices on the spine of the caterpillar, since the leaves cannot be centers of stars. The strategy for Minimizer is to respond with a star containing two vertices of the spine. This leaves one or two such caterpillars and some
isolated vertices. Minimizer chooses the response so that at all stages at most one caterpillar has a spine of odd length.

\[ \text{QED} \]

![Figure 3: A $P_3$-packable tree $E_6$ with smallest value of star-game for Minimizer responding](image)

Consider the stripe-game with Minimizer responding. If $G$ is a graph with $\mu(G) = m$, then by the same trivial argument as before the value of the game is at least $\lceil m/3 \rceil$. It seems unlikely that this bound is achievable in general, but we are unable to prove this. In general, we pose the question:

**Question 1** What is the best possible lower bound for the value of the stripe-game with Minimizer responding as a function of the $P_3$-packing number?

As in the case of the star-game, one can prove a better bound if the graph is a tree:

**Proposition 7** Consider the stripe-game with Minimizer responding. If $T$ is a tree with $\mu(T) = m$, then the value of the game is at least $\lceil m/2 \rceil$ and this is sharp.

**Proof.** For bound: consider an optimal $P_3$-packing $\mathcal{P}$ of $T$. As before we claim that Maximizer can ensure that the number of clean copies of $P_3$ decreases by at most 2 each round. To this end, Maximizer initiates on a vertex $v$ that is a leaf in the remaining forest. If $v$ is not in a clean copy of $P_3$, then the claim follows immediately. So assume $v$ is in a clean copy $Q$ of $P_3$. Then when Minimizer responds, their only option for the neighbor of $v$ is its neighbor in $Q$, and so the stripe taken can overlap at most two clean copies of $P_3$ from $\mathcal{P}$.

Optimality: Consider the caterpillar in Figure 3. When $m$ is even, no matter where Maximizer initiates, Minimizer can respond by using two vertices of the spine and leaving the remaining vertices of the spine to induce paths each with an even number of vertices. The argument when $m$ is odd is similar.  

\[ \text{QED} \]
4 Fully Packed Graphs

In the previous section we considered graphs where the value of the game is small. At the other end of the spectrum are those graphs where the number of copies taken is the largest it can be. For a particular game, we say a graph is **fully packed** if the result of the game on that graph is that all vertices are taken.

If a graph has a $K_3$-packing (meaning one can partition the vertex set into triples such that each induces a triangle), then it is immediate that Maximizer as responder can get all the vertices, whether it is the star- or stripe-game. For example, it is easy to build cubic graphs that have a $K_3$-packing (just start with a collection of triangles and add a perfect matching).

So we ask for trees: which trees are fully packed? In the following we use the fact that a disconnected graph is fully packed if and only if each component is fully packed.

4.1 Fully packed trees with Maximizer as responder

Let $\mathcal{D}$ be the family of forests defined as follows. Take some number of disjoint 2-stars and then add edges between their centers without creating a cycle. (One might call each component the double-corona of a tree.) This includes for example the caterpillar shown in Figure 3.

**Proposition 8** Consider the star-game with Maximizer responding. Then the trees in $\mathcal{D}$ are the fully packed trees.

**Proof.** The graphs in $\mathcal{D}$ are fully packed, since Minimizer is forced each time to initiate on a center vertex $v$, and Maximizer can respond by taking $v$ and its two leaves, leaving a member of $\mathcal{D}$.

We argue that these are the only fully packed trees. Consider a fully packed tree $T$ with an initiation by Minimizer on vertex $a$ and Maximizer’s chosen response using vertices $b$ and $c$. Suppose that $b$ is a valid initiation vertex for Minimizer, and Maximizer’s response to that would include the new vertex $x$. Let $C_x$ be the component of $T - \{a, b, c\}$ containing $x$; this is by assumption
$P_3$-packable and so has order a multiple of 3. But if Minimizer initiates on $b$ and Maximizer responds using $x$, then this uses one vertex from $C_x$ and so what remains does not have order a multiple of 3, and so the tree is not fully packed. That is, any potential initial vertex $v$ must have at least two leaf neighbors. But if any vertex has more than two leaf neighbors, then the graph is not $P_3$-packable; so it follows that $v$ has exactly two leaf neighbors.

The result then follows by induction, since what remains must be fully packed. (Note that the above argument shows that all of $v$’s other neighbors must have two leaf neighbors, since they too are potential initial vertices.) QED

**Proposition 9** Consider the stripe-game with Maximizer responding. Then $P_3$ itself is the only fully packed tree.

**Proof.** If a tree $T$ has more than three vertices, then it contains a non-leaf vertex $v$ that has exactly one non-leaf neighbor. Minimizer initiates on $v$, so that $v$’s leaf neighbors immediately become isolated. QED

### 4.2 Fully packed trees with Minimizer as responder

Let $\mathcal{E}$ be the family of forests defined as follows. Start with some number of $P_3$’s. Repeatedly add a $P_3$ and add at most one edge between it and each existing component, except no edge is added incident with the central vertex of the new $P_3$. A member of $\mathcal{E}$ is drawn in Figure 4.

![Figure 4: A fully packed tree for star-game with Minimizer responding](image)

**Proposition 10** Consider the star-game with Minimizer responding. Then the trees in $\mathcal{E}$ are the fully packed trees.

**Proof.** The graphs in $\mathcal{E}$ are fully packed, since Maximizer can initiate on the central vertex $v$ of the final added $P_3$ and then use recursion. (Minimizer is never given a choice.)
We argue that these are the only fully packed trees. Consider a fully packed tree \( T \) with initiation by Maximizer on a vertex \( a \) and assume one possible response is the star \( bac \). Suppose \( a \) has degree more than 2 and let \( x \) be one of \( a \)'s remaining neighbors. Since the component \( C_x \) of \( T - \{a, b, c\} \) containing \( x \) is \( P_3 \)-packable, it has order a multiple of 3. If instead Minimizer plays \( bax \), then we have still isolated \( C_x \) but removed one vertex from it, so it does not have order a multiple of 3 anymore, a contradiction. Thus we have shown that Maximizer must initiate on a vertex of degree 2. After removal of the star, apply induction. QED

Let \( F \) be the family of forests defined as follows. Start with nothing. Repeat-edly add a \( P_3 \) with a designated end-vertex \( v \), and join \( v \) to at most one vertex in each existing component. An example is shown in Figure 5, where the designated end-vertices are numbered in order of creation.

![Figure 5: A fully packed tree for stripe-game with Minimizer responding](image)

**Proposition 11** Consider the stripe-game with Minimizer responding. Then the trees in \( F \) are the fully packed trees.

**Proof.** The graphs in \( F \) are fully packed, since Maximizer can initiate on the end-vertex of the final added \( P_3 \) that is not \( v \), and then use induction/recursion. (Minimizer is never given a choice.)

We argue that these are the only fully packed trees. Consider a fully packed tree \( T \) with initiation by Maximizer on a vertex \( a \) and assume one possible response by Minimizer is the stripe \( abc \). Suppose \( a \) is not an end-vertex and let \( x \) be one of \( a \)'s remaining neighbors. Let \( C_x \) be the component of \( T - \{a, b, c\} \) containing \( x \). Since it is \( P_3 \)-packable, \( C_x \) has order a multiple of 3; further \( x \) has another neighbor, say \( y \). Consider the result if Minimizer responds by taking \( axy \). This removes two of the vertices from \( C_x \); and so what remains of \( C_x \), does not have order a multiple of 3, a contradiction. That is, \( a \) must be a leaf.
Suppose now that \( b \) has degree more than 2; say with another neighbor \( z \). Since the component \( D_y \) of \( T - \{a, b, c\} \) containing \( z \) is \( P_3 \)-packable, it has order a multiple of 3. But if the opening move is \( abz \), then we have still isolated \( D_y \) but removed one vertex from it, so it does not have order a multiple of 3 anymore, a contradiction.

Thus we have shown that Maximizer must initiate on a vertex \( a \) such that \( a \) is an end-vertex and its neighbor \( b \) has degree 2. After removal of \( abc \), apply induction. QED

4.3 Maximal outerplanar graphs

Recall that a maximal outerplanar graph (MOP) is created by taking a cycle and adding noncrossing chords until their addition is impossible. These graphs are a subset of the 2-trees. We consider which such graphs are fully packed.

The triangle \( K_3 \) is always fully packed. There are three MOPs of order 6: the fan, snake, and sun, drawn here.

![Figure 6: The three MOPs on 6 vertices](image)

Perhaps surprisingly, which is fully packed does not depend on the graph but only on the game. Minimizer responding in the stripe-game can ensure the value is 1; In the other three games, each graph is fully packed. We omit the details. The imperfection generalizes:

**Proposition 12** Consider the stripe-game with Minimizer responding. Then the only fully packed maximal outerplanar graph is \( K_3 \).

**Proof.** Assume the graph is fully packed and Maximizer initiates at vertex \( v \). Suppose that \( v \) has degree more than 2 and let \( vw \) be a chord incident with \( v \).
Then removal of \{v, w\} separates the graph $G$ into two components; let $x$ be a neighbor of $w$ on the outer cycle, chosen in the component of order a multiple of 3 if there exists such a component. Then Minimizer plays the stripe $vwx$ and leaves neither component a multiple of 3, and hence not all vertices are eventually taken. That is, the graph is not fully packed.

So vertex $v$ has degree 2. Say its neighbors are $y_1$ and $y_2$ and we are not in $K_3$. Then $y_1$ and $y_2$ have another common neighbor, say $z$. The removal of stripe $vy_1z$ creates a component $C$ without $y_2$; the removal of stripe $xy_2z$ leaves a component consisting of all the vertices of $C$, together with $y_1$. So at least one of these removals produces a component whose order is not a multiple of 3, which gives Minimizer a suitable response to avoid all vertices being taken. That is, the graph is not fully packed. QED

It is unclear what the fully packed MOPs look like for the other three games. We do note that the first part of the above proof carries over to the star-game on a MOP with Minimizer responding: to have a chance of a perfect outcome, Maximizer must initiate on a vertex of degree 2.

5 Some Grid-Like Graphs

5.1 Grids with two rows

Proposition 13 Consider the star-game. For $m \geq 2$ the value of the game on a $2 \times m$ grid is $\lceil m/2 \rceil$, regardless of which player has which role.

Proof. Think of the grid as 2 rows and $m$ columns. Consider first the game with Maximizer initiating.

Maximizer as initiator can ensure at least the desired quantity. Play top row first column, then top row third column, and so on. If $m$ is odd, add one final move of bottom row, last column. Each response is forced: each initiated vertex has exactly two neighbors at the time of being chosen. See Figure 7.
Minimizer as responder can ensure at most the desired quantity. Assume first that \( m \) is even. Minimizer will always use a vertical edge; further, if the star is initiated in column \( i \) then: if \( i \) is odd, they use a vertex in column \( i + 1 \); and if \( i \) is even, they use a vertex in column \( i - 1 \). Equivalently, Minimizer partitions the grid into \( 2 \times 2 \) subgrids, and responds to an initiation in some \( 2 \times 2 \) subgrid by staying within that subgrid. Note that the fourth vertex of the \( 2 \times 2 \) subgrid can never thereafter be chosen by the initiator.

If \( m \) is odd, Minimizer plays the same strategy where possible. Specifically, Minimizer tentatively partitions the grid into \( 2 \times 2 \) subgrids with one “floating” column in the last column. If Maximizer initiates in the last column, then Minimizer uses the vertical edge and one vertex of column \( m - 1 \), as forced. Mentally, Minimizer slides the floating column two to the left and continues the strategy. If initiator plays in the floating column a second time, then again Minimizer uses the vertical edge and the vertex to the left, and slides the floating column two to the left. Eventually the floating column will be surrounded by played \( 2 \times 2 \) subgrids. See Figure 8 for an example. Thus the number of moves is at most one more than the number of \( 2 \times 2 \) grids, which equals \( \lceil m/2 \rceil \).

The analysis of the game with the roles reversed is the same! Minimizer initiating can use the same strategy to end the game in \( \lceil m/2 \rceil \) moves. Maximizer responding can use the same strategy to ensure it lasts at least \( \lfloor m/2 \rfloor \) moves.

QED

**Proposition 14** Consider the stripe-game. For \( m \geq 2 \) the value of the game on a \( 2 \times m \) grid is \( \lceil m/2 \rceil \) if Maximizer responds and \( \lfloor m/2 \rfloor \) if Minimizer responds.

**Proof.** Minimizer as initiator can ensure at most \( \lceil m/2 \rceil \) moves as follows. Play in the top left corner. If Maximizer’s stripe is horizontal, then play in the bottom
row in the second column and repeat the strategy as if the first four columns are
gone. If Maximizer’s stripe uses a vertical edge, then repeat as if the first two
columns are gone. See Figure 9. Similarly, Maximizer as initiator can ensure at
least \( \lfloor m/2 \rfloor \) moves by the same strategy.

![Figure 9: Start of a stripe-game in the 2 × 9 grid](image)

Now, we argue that Minimizer as responder can ensure at most \( \lfloor m/2 \rfloor \) moves.
Their strategy is as follows. Number the columns 1, 2, up to \( m \). Every response,
the Minimizer:

> uses two vertices from an even-numbered column and one vertex from
> an odd-numbered column.

We claim Minimizer can always achieve this. The proof is by induction. Suppose
first that Maximizer initiates with \( v \) in an odd-numbered column \( C \). Consider
a column \( D \) next to \( C \). If both vertices in \( D \) are unused, then Minimizer
uses them. If not, then by the strategy, both vertices in \( D \) are used. But if that
happens for all columns next to \( C \), then \( v \) is not a valid start vertex. Suppose
second that Maximizer initiates with \( v \) in an even-numbered column \( D \). By the
strategy, the other vertex in \( D \) is unused. If Minimizer cannot respond by starting
with the vertical edge, then that means the neighboring odd-numbered columns
\( C \) each have a vertex taken (or don’t exist). It follows that the even-numbered
columns next over are, by the strategy, taken (or don’t exist); and so this \( v \) is
again not a valid move for Maximizer (instead \( v \) is the center of a star with at
most 3 leaves).

There are \( \lfloor m/2 \rfloor \) even-numbered columns. And so that is an upper bound on
the number of moves the Minimizer can be forced to make.

Maximizer as responder can ensure at least \( \lceil m/2 \rceil \) moves. This uses a similar
idea to above. Specifically:
every response uses two vertices from an odd-numbered column and one vertex from an even-numbered column.

As above, if such a response is not possible, then the initiator chose an invalid vertex. Further, we claim that if there is no valid vertex available for the initiator, then every odd-numbered column is full. The lower bound and the result follows. QED

5.2 Grids with three or more rows

Maximizer can do well sometimes on the grid with three rows.

**Proposition 15** For the star-game with Minimizer responding, the three-row grid is fully packed.

**Proof.** Assume we have a grid with three rows and with columns numbered from 1 up to \( m \). There are two distinct strategies based on the parity of \( m \).

**Even case:** Maximizer starts in the lower left corner. This gives Minimizer a forced move. Then, Maximizer plays the bottom vertex in the third column, forcing the Minimizer again. This continues with Maximizer playing the bottom vertex in column \( 2i + 1 \) for increasing \( i \). Thereafter, Maximizer plays in the top right corner, again forcing the Minimizer, and then moves left across the top row initiating in column \( 2i \) for decreasing \( i \).

**Odd case:** Here Maximizer starts with the middle vertex in the first column. If Minimizer responds within the column, then we are back in the even case. So without loss of generality assume that Minimizer responds by using the vertex in the second column and the vertex in the bottom left hand corner. Then Maximizer initiates in the top row second column. Minimizer has a forced response. Maximizer continues by playing in column \( 2i \) in the top row for increasing \( i \), followed by playing along the bottom row in column \( m - 2i \) for increasing \( i \). See Figure 10. Each response after the first move is forced. QED
It can be shown that for the stripe-game on a three-row grid with \(m\) columns, with Maximizer as initiator, the value is \(m\) if \(m \leq 3\), and \(m - 1\) otherwise. In contrast, the value of either game when Minimizer initiates is asymptotically at most \((1 - \varepsilon)m\) for some \(\varepsilon > 0\). We omit the proofs. (In particular we do not know the optimal \(\varepsilon\).)

For general grids, we note that initiator can use the same strategy as in Proposition 13 to achieve \(3/4\) the vertices being used. But it is unclear whether this is good, bad, or indifferent.

5.3 Rooks graphs with two rows

We consider rooks graph with two rows. That is, the graph \(R_m\) obtained by taking two cliques of size \(m\) and adding a perfect matching between them, which we call cross-edges. For simplicity, we restrict to \(m\) a multiple of 3.

Every maximal \(P_3\)-packing for such graph will leave at most 3 vertices; and if so, the residue will be 2 vertices from one clique and 1 from the other. Thus the value of the game is either \(m\) or \(m - 1\).

If Maximizer is the responder, then they can stay within the row, maintaining each row’s size a multiple of 3, and therefore always get every vertex. That is:

**Proposition 16** If Maximizer is responding, then for both games \(R_m\) is fully packed for all \(m\) a multiple of 3.

So we consider the game where Minimizer responds.

**Theorem 17** Consider the star-game with Minimizer responding. Then \(R_m\) is fully packed for all \(m\) a multiple of 3.
Proof. Maximizer initiates somewhere. (The graph is vertex-transitive.) There are two cases.

**Case 1: Minimizer stays within that row.** Then, Maximizer initiates at any vertex that does not have a cross-edge. This forces Minimizer to play within the row, and ensures that the number of vertices in each row remains a multiple of 3. If every vertex has a cross-edge, then we are back to a rooks graph, and can apply induction; otherwise Maximizer continues with a vertex that does not have a cross-edge.

**Case 2: Minimizer uses the cross-edge for the first star.** Say Maximizer initiated in the top row, so that two vertices were taken from the top row and one from the bottom row. Then Maximizer plays the vertex in the bottom row that has no cross-edge, to which Minimizer is forced to respond by taking three in the bottom row. Thereafter, Maximizer plays a vertex in the bottom row, and repeats so long as Minimizer stays within that row.

Eventually, since the number of unused vertices in the bottom row remains not a multiple of 3, Minimizer is forced to use the cross-edge. At that point, both rows have number of vertices left a multiple of 3; thus we are back in Case 1. QED

**Theorem 18** Consider the stripe-game, with Minimizer responding. Then $R_3$ is fully packed but $R_m$ for $m \geq 6$ is not.

Proof. It can easily be checked that $R_3$ is fully packed: whatever the first move, what is left is connected on 3 vertices and thus can be taken.

Consider $R_m$ for $m \geq 6$. Minimizer’s strategy will ensure that, until the very end, every vertex in the smaller row still has its cross-edge. Therefore, we can refer to the situation by just the counts of the two rows. We will use $(i, j)$, with $i \geq j$, to denote the situation where one row has $i$ vertices and one row has $j$ vertices. For the base of the induction, we need the case $(4, 2)$. For this, one can readily check that wherever Maximizer initiates, Minimizer can respond and disconnect the graph, thereby ending the game.

For $R_m$ the play starts at the case $(m, m)$. For the first move, Minimizer uses a cross-edge; so the case becomes $(m - 1, m - 2)$. We claim that Minimizer
can ensure the case \((m - 2, m - 4)\) next. For, if Maximizer chooses a vertex in the larger side, then Minimizer stays in the larger side, using up the vertex that has no cross-edge; and if Maximizer chooses a vertex in the smaller side, then Minimizer takes two vertices there and a cross-edge to the other row.

We claim that Minimizer can ensure the case \((m - 4, m - 5)\) next. For, if Maximizer initiates on the larger side, then Minimizer takes three vertices there; and if Maximizer initiates on the smaller side, then Minimizer immediately uses the cross-edge and then takes one of the (two) vertices without a cross-edge. By repeated application of the strategy, Minimizer can alternate between cases of the form \((x, x - 2)\) and \((y, y - 1)\) until they reach the case \((4, 2)\), which we saw is not fully packed. QED

6 The Unrooted \(P_3\)

There is also a version of the game where the packing subgraph has no root. We define the unrooted-\(P_3\)-game to be the game where responder need only choose a copy of \(P_3\) containing the designated vertex. We show that in some cases the value of the game is the same as in the rooted version, but in other cases it is different. In particular, for Maximizer responding, they can always produce a maximum \(P_3\)-packing.

6.1 Maximizer responding

**Theorem 19** Consider the unrooted-\(P_3\)-game with Maximizer responding played on graph \(G\). Then the value of the game is \(\mu(G)\).

**Proof.** Consider a maximum \(P_3\)-packing \(\mathcal{P}\) of \(G\). If Minimizer chooses a vertex \(u\) in \(\mathcal{P}\), then Maximizer responds with the associated copy of \(P_3\) in \(\mathcal{P}\) and repeats. If instead Minimizer chooses a vertex \(v\) outside \(\mathcal{P}\), then by requirement, the vertex \(v\) is in a copy \(Q\) of \(P_3\). The only way a problem could arise is if \(Q\) intersects two clean copies of \(P_3\) in \(\mathcal{P}\). But that implies that vertex \(v\) has an edge to some clean copy of \(P_3\) in \(\mathcal{P}\); and so Maximizer can use that edge and one edge from that clean copy to build a \(P_3\), thereby affecting only one clean copy of \(P_3\) in \(\mathcal{P}\). Repeat. QED
The above theorem generalizes Theorem 1 on the matcher game from [4]. Note that this pattern does not continue much further. In particular, the star $K_{1,3}$ does not have a similar result. For example, take three copies of $K_{1,3}$ and add a new vertex $v$ adjacent to one end-vertex from each copy. The resulting tree has three disjoint $K_{1,3}$’s, but Minimizer ends the game in one move by initiating on $v$.

6.2 Minimizer responding

When one changes to the unrooted game, this gives both players more options. We saw above that when Maximizer is responder, the added options to Minimizer do not help them. A similar result holds if we go from the stripe-game to the unrooted game with Minimizer as responder:

**Theorem 20** Consider Minimizer responding. The value of the unrooted-$P_3$-game is at most the value of the stripe-game.

**Proof.** Consider changing from the stripe-game to the unrooted game. Minimizer plays just as if it were the stripe-game. We argue that the new options do not help Maximizer. For, the only additional option they have is to play a vertex $v$ that is in a $P_3$ but is not the end of one. That is, the only additional initiation option they get is to play the center vertex of a star component; but that is equivalent to initiating at a leaf of the component, which they could do already. QED

In contrast, the star-game is incomparable with the unrooted-$P_3$-game. Consider, for example, the double corona of a complete graph (see Figure 2). Maximizer now can initiate on a leaf, and thereby ensure that approximately half the vertices are used. On the other hand, consider the tree shown in Figure 4. As we saw, Maximizer can obtain every vertex in the star-game by initiating on the central vertex $v$. In the unrooted $P_3$-game, however, Minimizer can respond differently and destroy the perfection.
7 Questions

We conclude with some questions for future study. Obviously, a natural direction is to replace $P_3$ by another required subgraph. For the games with $P_3$, it would be interesting to determine the value of the game on a general grid and a general rooks graph. Another question, is whether there is a $\varepsilon > 0$ such that all graphs of order $n$ a multiple of 3 with minimum degree at least $(1 - \varepsilon)n$ are fully packed. The analogous question for $K_2$ is also still open.

8 Acknowlegements

We would like to thank all the referees for their careful reading of the manuscript (despite our sloppiness at times) and their generous suggestions.

References


