DIMACS Technical Report 95-10
April 1995

A Winning Strategy for the Ramsey Graph Game

by

Aleksandar Pekeč[1,2]
Department of Mathematics
Rutgers University
New Brunswick, New Jersey 08903

[1] Affiliated Graduate Student Member, DIMACS Graduate Assistant
[2] The author gratefully acknowledges the support of the Air Force Office for Scientific Research under grant number F49620-93-1-0041 to Rutgers University.

DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

We consider a "Maker-Breaker" version of the Ramsey Graph Game, RG(n), and present a winning strategy for Maker requiring at most $(n-3)2^{n-1}+n+1$ moves. This is the fastest winning strategy known so far. We also demonstrate how the ideas presented can be used to develop winning strategies for some related combinatorial games.
Keywords: Combinatorial Games, Algorithms on Graphs, Ramsey Theory

1 Introduction

The Ramsey Graph Game, RG(n), on a complete graph on N vertices, K_N, is considered. Two players, Maker (red) and Breaker (blue) alternately color the edges of K_N. Maker is first to play, and the players color one edge per move. Maker wins the game if there is a red K_n. Breaker wins if there is no red K_n after all the $N(N-1)/2$ edges have been colored.

Let $R(n,n)$ denote the n-th Ramsey number, i.e., the smallest number R such that for every two-coloring of the edges of K_R there exsist a monochromatic $K_n \subset K_R$.

Maker has a simple winning strategy if $N \geq 2R(n,n)$. Note that K_N contains two disjoint copies of $K_{R(n,n)}$. In other words there are $K^1, K^2 \subset K_N$ such that $V(K^1) \cap V(K^2) = \emptyset$ and $K^1 \simeq K^2 \simeq K_{R(n,n)}$. Every time Breaker colors an edge from K^1 or K^2, Maker responds by coloring the corresponding edge in the other copy of $K_{R(n,n)}$ (K^2 or K^1, respectively). If this edge is already colored (it must be colored red), or if Breker didn't color an edge from $E(K^1) \cup E(K^2)$, then Maker colors an arbitrary edge from K^1. Maker can claim a victory after K^1 has been colored. If there is no red $K_n \subset K^1 \subset K_N$ then, by the Ramsey Theorem, there is a blue $K_n \subset K^1$. Hence, the copy of the blue K_n is red in K^2, i.e., there is a red $K_n \subset K^2 \subset K_N$. This pairing strategy is called "the mirror-image strategy" ([4, 7]).

If $N \geq R(n,n)$ then Maker is guaranteed to have a winning strategy by the strategy stealing argument. More precisely, if Maker has no winning strategy, then Breaker has a strategy to prevent red K_n. However, any such strategy will result in building a blue K_n (Since $N \geq R(n,n)$, there must be a monochromatic copy of K_n). Maker can steal Breaker's strategy by playing a random first move and then pretend to be a second player following Breaker's strategy which will enable him to build a K_n made of Maker's edges only. It should be noted that Maker's (random) first move could only help since an additional Maker's edge can only be an advantage in building a K_n. Although Maker is guaranteed to have a winning strategy this argument is nonconstructive.

The next natural question is can Maker win faster, i.e., can Maker win if $N < R(n,n)$ and, ultimately, what is the minimal number of moves needed for Maker to win the game. Using the weight function method, Beck [3] showed that Maker can win if $N > (2+\epsilon)^n$ for n sufficiently large. By setting up an appropriate weight function, a weight is assigned to every $K_n \subset K_N$ and Maker picks an edge (to be colored red) in order to maximize the total weight (i.e. sum of the weights of all K_n's). This winning strategy will guarantee that K_N will contain not only a copy of red K_n but also a copy of any 2-edge-colored K_n. However, Maker needs to play all the $\lceil N(N-1)/4 \rceil$ moves to assure the exsitence of a red K_n only.

On the other hand, by the Theorem of Erdős and Selfridge [9] (which is essentially the weight function method), Breaker can win the game if $N < n2^{(n-1)/2}$.

The winning strategy requiring $(n-3)2^{n-1}+n+1$ moves is presented in this paper. As it will be shown later, Maker can apply this strategy whenever $N \geq (n-2)2^{n-1}+2$. It is important to note that the number of moves is independent of N.

2 Preliminary Remarks

Let $G_M \subset K_N$ denote a graph induced by the red edges. In this notation, Maker's goal is to construct G_M such that $K_n \subset G_M$.

A vertex $v \in V(K_N)$ is **introduced** by Maker in a particular move if it wasn't in G_M before that move and it becomes a vertex in G_M after that move . Note that no vertex is introduced by Maker in the last $n-2$ moves of any winning strategy (because every vertex in K_n has degree $n-1$).

In any winning strategy, one move before Maker's winning move G_M contains at least two $K_n - e$ (where $K_n - e$ denotes a complete graph on n vertices without an arbitrary uncolored edge e). This is a winning trap since Breaker can't prevent Maker to extend one of those graphs to $K_n \subset G_M$ (We assume that there are at least two e's among all $K_n - e \subset G_M$). Obviously, the smallest G_M containing a winning trap consists of two K_n's each missing an edge and intersecting in K_{n-1}. Let us call such a graph $W_1(n)$. Maker can win in two moves if G_M contains at least two $W_1(n) - e$. The smallest such G_M consists of two copies of

$W_1(n) - e$ intersecting in K_{n-2}. Let us call such a graph $W_2(n)$. For every $k = 2, \ldots n-2$, $W_k(n)$ can be defined inductively as two copies of $W_{k-1}(n) - e$ intersecting in K_{n-k}.

Let us define $S(2) = K_1$, and $S(k+1)$ to be the graph obtained from $S(k)$ by introducing two new vertices for each $K_{k-1} \subset S(k)$ and connecting them with every vertex of the corresponding K_{k-1}.

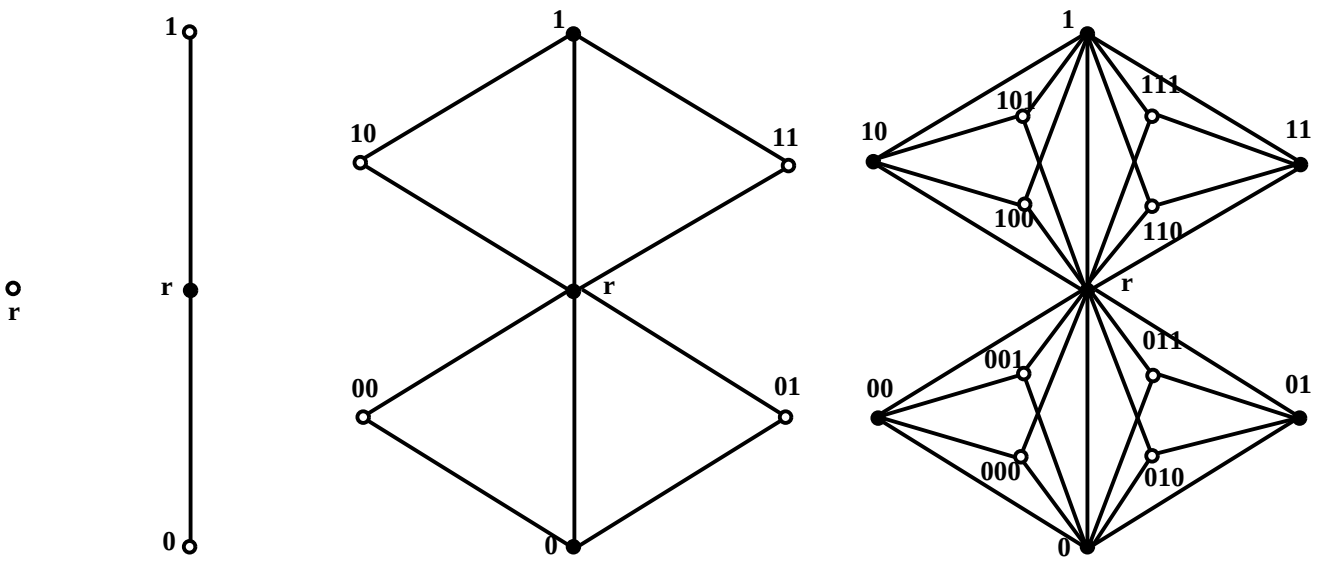

Figure 1: S(2), S(3), S(4), S(5)

$S(k)$ is a union of 2^{k-2} K_{k-1}'s which all intersect in a single vertex r which will be called a **root**. Moreover, the K_{k-1}'s can be naturally paired so that each pair intersects in a K_{k-2}, and these pairs can be paired so that each pair of pairs intersects in a K_{k-3}. This procedure can be continued inductively: the last pairing will result in all K_{k-1}'s intersecting in a common K_1 (the root).

Note that if $S(n) \subset G_M$ then Maker can win in exactly $n-1$ moves. One way to see that is to observe that $S(n) + \{r, w\} = W_{n-2}(n)$ (where w is a newly introduced vertex from $V(K_N) - V(S(n))$ and the K_n's from the definition of $W_{n-2}(n)$ are exactly all $K_{n-1} + \{r, w\}$). What is really going on is that Maker has a pairing strategy on $S(n) + \{r, w\}$. In order to win the game, Maker needs to connect w to $n-2$ vertices of some K_{n-1} (since w is connected to r which is a vertex in every K_{n-1}). The most Breaker can do is to connect w with a vertex in one of the K_{n-1}'s. Maker will restrict his moves on the other half of the K_{n-1}'s according to the natural pairing, and will connect w to other vertex of the K_2 ($r \in K_2$) shared by all K_{n-1}'s in that group. The same strategy continues, ultimately leading to a winning trap where w will be connected (in G_M) to every vertex of K_{n-2} shared by two $K_{n-1} \subset G_M$. This is exactly $W_1(n)$.

In order to win the game in this way, Maker needs to build a $S(n) \subset G_M$. In a winning strategy which will be described, Maker will either win the game or have $G_M = S(n)$ after $(n-3)2^{n-1} + 2$ moves. This goal seems to be hopeless since Breaker can easily prevent Maker from expanding G_M symmetrically (i.e. building two K_k's on each K_{k-1}). In order to do that, Breaker will be forced to let Maker build large complete graphs on the part of the maximal $S(i) \subset G_M$ (Note that $S(2)$ and $S(3)$ can be built trivially). But Breaker cannot allow $K_n \subset G_M$ and this will force him to let Maker build $G_M = S(n)$ which is nothing but a winning trap in $n-1$ moves.

In the next section this winning strategy will be formally defined.

3 A Winning Strategy

Observation 1 *For every $n \geq 2$,*

1. *$\omega(S(n)) = n-1$ and there are exactly 2^{n-2} cliques of size ω in $S(n)$.*
2. *$|V(S(n))| = 2^{n-1} - 1$.*
3. *$|E(S(n))| = (n-3)2^{n-1} + 2$.*

Proof: By induction. Obviously, the statements are true for $n = 2$. Suppose the statements are true for $n \leq k$. Then $S(k+1)$ is obtained from $S(k)$ by adding two vertices for each K_{k-1} and connecting both vertices to every vertex of the corresponding K_{k-1}. Therefore, $\omega(S(k+1)) = \omega(S(k)) + 1$ and the number of maximal cliques is doubled, i.e., there are $2 \cdot 2^{k-2} = 2^{k-1}$ K_k's in $S(k+1)$ and we have:
$|V(S(k+1))| = |V(S(k))| + 2 \cdot 2^{k-2} = 2^{k-1} - 1 + 2^{k-1} = 2^k - 1$.
Also, $|E(S(k+1))| = |E(S(k))| + 2 \cdot (k-1) \cdot 2^{k-2} = (k-3)2^{k-1} + 2 + (k-1)2^{k-1} = 2(k-2)2^{k-1} + 2 = ((k+1)-3)2^{(k+1)-1} + 2$. ∎

There is a natural way to label vertices of $S(n)$ with binary strings of length $< n$. We start labeling the only vertex of $S(2) = K_1$ with r (root) which will correspond to the empty binary string. Two new vertices of $S(3)$ are labeled by two strings of length one: 0 and 1. We continue labeling new vertices inductively: every vertex v from $V(S(k+1)) - V(S(k))$ will be labeled with a binary string of length $k-1$ in a way that v is adjacent to every vertex $w \in V(S(k))$ labeled with the substring of first i ($0 \leq i \leq k-2$) digits of the v's label. In this way every maximal clique (K_{k-1}) in $S(k)$ will have vertices with labels of all lengths from 0 to $k-2$, and the label of length i will be an extension of the label of length $i-1$ (there are two possible extensions since the last digit can be 0 or 1). Every maximal clique in $S(k)$ corresponds to two vertices from $V(S(k+1)) - V(S(k))$ and those two vertices are labeled with the longest label among all the labels of the vertices from the clique and extended with 0 for one vertex and with 1 for the other vertex. More formally:

1. Label the unique vertex from $S(2)$ with the empty binary string (r).
2. Label the vertices from $V(S(3)) - V(S(2))$ with binary strings of length 1: 0 and 1.
3. For $k := 3, 4 \ldots$ do:
   Label the vertices from $V(S(k+1)) - V(S(k))$: For every maximal clique in $S(k)$ find the vertex whose label l is a binary string of length $k-2$, extend that string by adding 0 or 1 at the end of the string (making it a string of length $k-1$) and label two new vertices corresponding to this clique with those two extended strings ($l0$ and $l1$).

Observation 2 *Every vertex of $S(n)$ is labeled with a binary string of length $\leq n-2$, every two vertices have different labels, and every maximal clique has a vertex labeled with a binary string of length $n-2$. Moreover, every binary string of length $\leq n-2$ is the label for some vertex of $S(n)$.*

Proof: By induction. The statement is true for $n = 2$ and $n = 3$. Suppose that the statement is true for $n = k$. Note that every vertex from $V(S(k+1)) - V(S(k))$ will be labeled with a string of length $k-1$. Since every two vertices in $S(k)$ have different labels, 2^{k-1} vertices from $V(S(k+1)) - V(S(k))$ will receive different labels of length $k-1$. Obviously, every binary string of length $k-1$ will be used as a label. ∎

In order to describe the winning strategy for the RG(n) game, let us introduce a game played on the full rooted binary tree with $n-1$ levels, $B(n)$.

The full rooted binary tree with $n-1$ levels is a digraph whose vertices are binary strings of length $\leq n-1$ and (l, s) is an arc in $A(B(n))$ if and only if l is the string obtained from s by erasing the last digit of s (i.e., $s = l0$ or $s = l1$). The empty string is called the root (r), and all the strings of length k are vertices on the level k. Vertices on the level $n-1$ are called leafs. For every vertex v, neighbors of v are defined to be $N(v) = \{w \in V(B(n)) : (v, w) \in A(B(n))\}$. Obviously, every vertex which is not a leaf has exactly two neighbors. It should be noted that $S(n+1)$ is isomorphic to the graph whose vertices are $V(T(n))$ and (l,s)

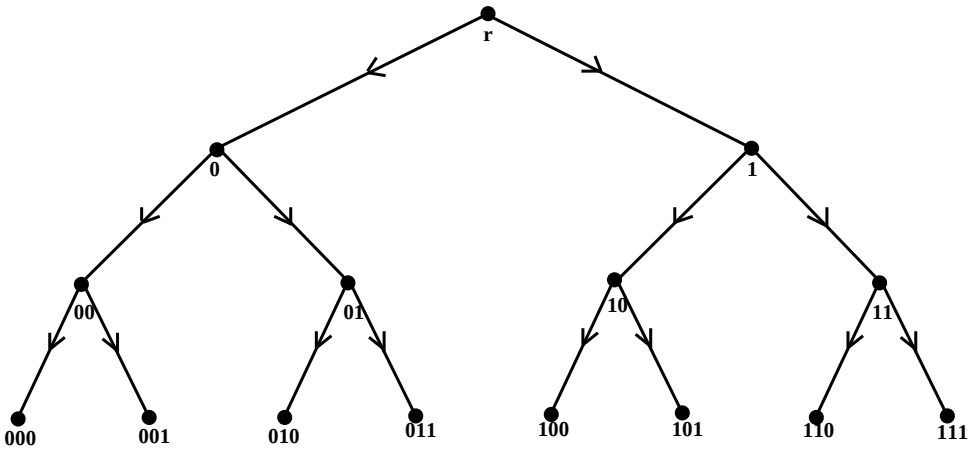

Figure 2: B(4)

is an edge in the graph if and only if in $B(n)$ there exists a directed path from s to l or a directed path from l to s.

The Leaf Game, LG(n), is a game played on the $B(n)$ by two players: Maker and Breaker. Maker wants to capture a leaf and Breaker wants to prevent him. At the beginning of the game the set of captured vertices is $C = \{\mathbf{root}\}$ and the position of the Maker is $p_M = \mathbf{root}$. In every move Maker moves down the tree through unblocked vertices and Breaker blocks the vertices. Breaker can block only the captured vertices. Every time Maker arrives at the vertex which is still not captured, that vertex becomes captured and the game is reset: Maker returns to the root and no vertex is blocked (the only change is that C got a new element). More precisely:

Leaf Game (n)

1. $C := \{\mathbf{root}\}$.
2. RESET. Let $p_M = \mathbf{root}$, $\mathbf{Trace} = \{\mathbf{root}\}$, $\mathbf{Blocked} = \emptyset$.
3. BREAKER'S MOVE. Breaker can
   (a) Pick $v \in C - (\mathbf{Trace} \bigcup \mathbf{Blocked})$ and set $\mathbf{Blocked} := \mathbf{Blocked} \bigcup \{v\}$.
   or
   (b) Do nothing.
4. MAKER'S MOVE. Maker picks $w \in N(p_M) - \mathbf{Blocked}$.
   (a) If there is no such w (i.e., $N(p_M) \subset \mathbf{Blocked}$) then BREAKER WINS
   (b) If $w \in C$ then set $p_M := w, \mathbf{Trace} := \mathbf{Trace} \bigcup \{w\}$ and go to Step 3 (Breaker's Move).
   (c) If $N(w) = \emptyset$ (w is a leaf) then MAKER WINS.
   (d) If $w \notin C$ then set $C := C \bigcup \{w\}$ and go to Step 2 (Reset).

Obviously, in the first move Breaker can't pick any vertex and Maker captures a new vertex and the game enters the reset mode prior to the second move. The larger the set C, the closer Maker will be to win the game but Breaker will have more choices for blocking.

Theorem 3 *Maker wins the LG(n) game in at most* $(n-3)2^{n-1} + n + 1$ *moves.*

Proof: Breaker can win only if $N(p_M) \subset$ **Blocked**. If this will never be the case, Maker will eventually win since $T(n)$ is acyclic. Suppose Maker follows the following strategy:

If in the last move Breaker picked v then Maker picks w which is not on the unique directed (p_M, v)-path. If there is more than one choice for w (e.g., Breaker did nothing in the last move or v can't be reached from p_M) then Maker can pick w arbitrarily among the candidates for w (To make the strategy deterministic we can set $w = p_M 0$).

By following this strategy Maker guarantees that there are no vertices from **Blocked** which can be reached from p_M before any of Breaker's moves, and that $N(p_M) -$ **Blocked** $\neq \emptyset$ before any of Maker's moves (i.e., this is a winning strategy). We prove this by induction on the number of moves after the reset mode. Before the first move we have **Blocked** $= \emptyset$ and the statement trivially holds. Suppose the statement is true before the Breaker's k-th move. After Breaker makes his k-th move, there is at most one vertex from **Blocked** reachable from p_M (namely: the v picked in the Breaker's k-th move) and the second part of the statement holds. Therefore, Maker has a choice for w, and no vertices from **Blocked** are reachable from w since any vertex reachable from w is reachable from p_M and w is not on the unique (p_M, v)-path (if such a path exsists at all). Hence, **Blocked** $= \emptyset$ before Breaker's $(k+1)$-st move.

It remains to show that the game is over in at most $(n-3)2^{n-1} + n + 1$ moves. In the worst case, Maker will capture every vertex which is not a leaf before winning the game by capturing the first leaf (Actually, Breaker can force Maker into this situation). In order to capture a vertex on the level k, Maker will need k moves. Therefore, Maker will need

$$(n-1) + \sum_{k=0}^{n-2} k2^k = (n-3)2^{n-1} + n + 1$$

moves. ∎

Corollary 4 *Maker can't win the LG(n) game in less than* $(n-3)2^{n-1} + n + 1$ *moves.*

Proof: It suffices to show that Breaker has a strategy which will force Maker to capture every vertex on level $n-2$ before capturing the first leaf (in order to capture a vertex on level $n-2$, Maker needs to capture all the vertices on the unique path from r to that vertex, and every vertex on level $k < n-2$ is on such a path).

Suppose that Maker has a strategy with ehich he can win without capturing all the vertices on the level $n-2$. Let us consider the game after the last reset (Maker is going to capture a leaf). There exists a vertex l on the level $1 \leq k \leq n-2$ which is not in the set C and Maker will not capture it at all (since that was the last reset and he will capture a leaf regardless of Breaker's moves). Suppose l is such a vertex with minimal k. Consider the unique (r,l)-path P: $r = x_0, x_1, x_2, \ldots, x_{k-1}, l$. For every $i = 1, \ldots, k$, let $y_i \in N(x_{i-1}) - P$ denote the unique neighbor of x_{i-1} not in P. Note that y_i is on level i. By the choice of l; $x_{i-1}, y_i \in C$ for every $1 \leq i \leq k-1$ ($x_{k-1} \in C$ also). In the i-th move after the last reset, Breaker will pick y_i and this will force Maker to pick x_i. Thus, after $k-1$ moves, $p_M = x_{k-1}$ and Breaker will pick y_k if $y_k \in C$ thereby forcing Maker to pick l. If $y_k \notin C$ then Breaker will do nothing and Maker will be forced to pick either y_k or l. In any case Maker will capture new vertex on the level $k < n-1$ and reset the game once more, a contradiction ∎

In order to win the RG(n) game, Maker can apply the winnning strategy for the $LG(n)$ game and relate it to the RG(n) game as follows:

1. ($C = \{$**root**$\}$) Before start of the game, Maker will pick $r \in V(K_N)$, label it **root**, and set $V(G_M) =$ {**root**}.

2. (Reset) Maker will pick a vertex $u \in V(K_N) - V(G_M)$ such that there is no blue edge connecting u with G_M. Maker colors {r,u} red in the RG(n) game.

3. (Breaker's move)

   (a) If Breaker colors $\{u, v\}$ blue for some v already labeled ($v \in V(G_M)$), that is considered as Breaker picking $v \in V((B(n))$ and making the play of type 3(a) in the LG(n) game.

   (b) If Breaker colors any other edge of K_N, that is considered as Breaker doing nothing in the LG(n) game (making a play of type 3(b)).

4. (Maker's move)

   (a) This is impossible if Maker follows a winning strategy for the LG(n) game.

   (b) Maker colors $\{u, w\}$ red in the RG(n) game.

   (c) Maker colors $\{u, w\}$ red in the RG(n) game and claims $K_n \subset G_M$.

   (d) Maker labels u with the string $w \in V(B(n))$.

It should be noted that Maker does not play a move in the case 4(d). However, this case is always immediately followed by the reset mode and Maker plays his move each time in the reset mode. Maker also plays the first move in the game by playing his move in the initial reset mode.

Every vertex except u in G_M is labeled, and Maker connects u with labeled vertices in G_M trying to connect u to a clique in G_M which is as large as possible. When this attempt is blocked (Maker captured new vertex in the LG(n) game), Maker labels vertex u (which is now part of some clique in G_M) and introduces a new vertex u (Reset mode).

The only way for Maker to get stuck is if there is no choice for new vertex u. However, if N is sufficiently large this will not be a problem.

Theorem 5 *Let n be given and $N \geq (n-2)2^{n-1}+2$. Then Maker wins the RG(n) game in at most $(n-3)2^{n-1}+n+1$ moves.*

Proof: We will first show that N is large enough for Maker to introduce a new vertex u whenever the LG(n) game is in the reset mode. After that we will prove that captured leaf in the LG(n)game means that $K_n \subset G_M$.

First note that the last time when the LG(n) game is in the reset mode is after at most $\sum_{k=0}^{n-2} k2^k = (n-3)2^{n-1}+2$ moves. The worst case can happen if Maker was forced to capture every vertex on the level $n-1$. At that point, $|C| = 2^{n-1}-1$. If some vertex v is not a candidate for u this means that either $v \in V(G_M)$ or there exists $m \in V(G_M)$ such that $\{v, m\}$ is a blue edge. Therefore, before the LG(n) game is in the reset mode for the last time, at most $(n-3)2^{n-1}+2$ vertices v from $V(K_N) - V(G_M)$ cannot be candidates for u (since every such v must be the endpoint of at least one blue edge). Hence, if $N > (n-3)2^{n-1}+2+2^{n-1}-1$ there will still be a candidate for u (when the LG(n) game is in the reset mode for the last time $|V(G_M)| = |C|$).

It remains to show that if there is a leaf $l \in C$ then $K_n \subset G_M$. Since $l \in C$, there exists a directed $(\mathbf{root}, l)$-path of vertices from C:

$$P: \quad \mathbf{root} = v_1, v_2, \ldots, v_n = l.$$

Note that v_i is connected to v_j in G_M for every $i > j$. Therefore, $G_M(v_1, \ldots, v_n) = K_n$. ∎

As it was noted in Corollary 4, Breaker can force Maker to play exactly $(n-3)2^{n-1}+n+1$ moves if Maker is applying the winning strategy described. On the other hand, more careful analysis might reduce the lower bound for N.

Labeling of the vertices may be used to define an orientation of the edges in G_M: If $\{u, w\}$ is to be colored red in the winning strategy (note that u has no label at that point), orient $\{u, w\}$ from w to u (making it (w, u)). In this way, Maker will orient G_M and have a transitive red K_n. Therefore, Maker has a winning strategy for the Transitive Tournament Game, TT(n). In the TT(n) game players alternately orient the edges and maker wins if there is a transitive tournament on n vertices. Moreover, Maker can claim a red transitive tournament, i.e., a transitive tournament made of edges oriented by Maker.

Corollary 6 *Let n be given and $N \geq (n-2)2^{n-1}+2$. Then Maker wins the TT(n) game in at most $(n-3)2^{n-1}+n+1$ moves.*

4 Generalizations and Concluding Remarks

A straightforward generalization of the RG(n) game is the biased Ramsey Graph Game, $RG_b(n)$, where Maker colors one edge per move while Breaker colors b edges per move. The winning strategy from the unbiased game can be easily modified for this case. The idea is that Maker needs $b+1$ $(K_n - e)$'s for the winning trap. The associated game, $LG_b(n)$, is the game played on the full rooted $(b+1)$-ary tree with $n-1$ levels (the vertices are strings of length $\leq n-1$ made of b different digits) and Breaker can block up to b vertices from $C - (\mathbf{Blocked} \bigcup \mathbf{Trace})$ per move. The following results are straightforward generalizations of the results from the previous section: Let

$$K = \sum_{k=0}^{n-2} k(b+1)^k = \frac{b+1}{b^2}((n-2)(b+1)^{n-1} - (n-1)(b+1)^{n-2} + 1).$$

Theorem 7 *Maker can win the $LG_b(n)$ game in at most $K+n-1$ moves. Conversly, Maker can't win in less than $K+n-1$ moves.*

Theorem 8 *Let n and b be given. If $N > K + ((b+1)^{n-1}-1)/b$ then Maker can win the $RG_b(n)$ game in at most $K+n+1$ moves.*

Maker also has a winning strategy for the biased version of the Transitive Tournament Game, $TT_b(n)$, where Brekaer orients b edges per move:

Corollary 9 *Let n and b be given. If $N > K + ((b+1)^{n-1}-1)/b$ then Maker can win the $TT_b(n)$ game in at most $K+n+1$ moves.*

Yet another generalization of the (biased and unbiased) Ramsey Graph Game is the game where Breaker can postpone his move, i.e., Breaker can color as many edges as he wants, provided that the total number of blue edges before before Maker's $(k+1)$-th move is no more than kb. The strategy from the last section is not a winning strategy in this case.

As it was mentioned before, the powerful weight function method gives a winning strategy if Maker's goal is to build any two-edge-colored K_n where colors are prescribed in advance. It seems that there is no chance for any modification of the strategy from the last section to guarantee Maker's victory in this game. The main reason for such a pesimistic view is the conjecture that the number of moves Maker needs to win this game depends on N.

The most interesting generalization of the RG(n) game is the Competitive Ramsey Graph Game, CRG(n), where both Maker and Breaker attempt to build K_n from their edges and the winner is the player who builds K_n first (second player becomes maker also). The Ramsey Theorem and the strategy stealing argument guarantee the existence of the winning strategy for the first player if $N \geq R(n,n)$. It is not known if the number of moves Maker needs to win the game is indepedent of N. Clearly, for $n=3$ Maker trivially wins in at most 4 moves, but when $n=4$ situation becomes more complicated. There are some claims that 15 moves suffice. Nothing is known for $n \geq 5$. One might hope that Maker can modify the winning strategy for the RG(n) game by carefully blocking Breaker's threats and use the rest of his moves to build his K_n. The main problem here is that it is much easier to build a trap which doesn't win the game but forces other player to defensive moves only as long as there are vertices which are not in G_M nor in G_B. More precisely, the subgraph of $S(n)$ defined by the root and all the vertices with the first digit 0 is an example of such a trap. Therefore, if Maker is blindly following the winning strategy for the RG(n) game, Breaker can build such a trap much faster. Still, it seems that if the number of moves needed to win the CRG(n) game is independent of N, then one should expect that some modification of the strategy from the last section will be the winning strategy for the CRG(n) game.

It would be interesting to see if the strategy described here could be generalized for the Generalized Ramsey Game where Maker and Breaker occupy k-set in every move and Maker wants to obtain an n-set whose all k-sets are occupied by Maker.

The main idea of the presented strategy is to analyze the winning traps few moves before the end of the game. In the case of the RG(n) game, we have shown that $S(n)$ is a winning trap where Maker wins in exactly $n-1$ moves. On the other hand, Maker can build such a trap. One might conjecture that this is the fastest winning strategy since $S(n)$ is in some sense the smallest winning trap in $n-1$ moves. More precisely, $S(n)$ is the smallest winning trap where Maker wins in exactly $n-1$ moves. Note that $|V(S(n))| = 2^{n-1}-1$ and, if $S(n) = G_M$, Maker needs to introduce an additional vertex to win the RG(n) game.

Conjecture: Let $N < 2^{n-1}$. Then Breaker has a winning strategy for the RG(n) game.

If the conjectrue holds, Breaker would have at least a drawing strategy for the CRG(n) game. Since Maker can't lose the CRG(n) game (the strategy stealing argument), the game will end in a draw. Therefore, if the conjecture holds, then $R(n,n) \geq 2^{n-1}$.

The ideas presented might be used in analysis of other types of combinatorial games since the idea of the winning trap, doubling the trap, and building a trap are relevant to variety of combinatorial games. It seems that the main reason why these ideas had been easily applied to the RG(n) game is the high level of symmetry inherent in this game.

Acknowledgement

The author gratefully acknowledges the support of the Air Force Office for Scientific Research under grant number F49620-93-1-0041 to Rutgers University.

References

[1] J. Baumgartner, F. Galvin, R. Laver, and R. McKenzie, Game theoretic versions of partition relations. *Colloq. Math. Soc. János Bolyai "Infinite and Finite Sets", Keszthely, Hungary* (1973), 131-135

[2] J. Beck, On positional games. *J. Comb. Theory A* **30**(1981), 117-133

[3] J. Beck, Van der Waerden and Ramsey type games. *Combinatorica* **1**(1981), 103-116

[4] J. Beck, Combinatorial Games. *Lecture Notes, Rutgers University* (1993)

[5] J. Beck, Games, Randomness and Algorithms. *manuscript* (1994)

[6] J. Beck and L. Csirmaz, Variations on a game. *J. Comb. Theory A* **33**(1982), 297-315

[7] E.R. Berlekamp, J.H. Conway, and R.K. Guy, Winning ways for your mathematical plays. *Academic Press, NY* (1982)

[8] V. Chvátal and P. Erdős, Biased positional games. *Annals of Discrete Math.* **2**(1978), 221-228

[9] P. Erdős and J. Selfridge, On a combinatorial game. *J. Comb. Theory A* **14**(1973), 298-301

[10] R.L. Graham, B.L. Rothschild, and J.H. Spencer, Ramsey Theory. *Wiley, NY* (1990)