On the Workday Number for Finite Multigraphs in a Variation of Cops and Robbers

Eric Schneider

2323 Frist Center, Princeton University, Princeton, NJ 08544

Abstract

The classical mathematical problem of "Cops and Robbers on a graph"' has applications both in pure mathematics and in modeling real world problems such as wildfires and disease control. We investigate a variant of this problem in which the Cops choose to inspect any number of vertices on the graph, but they lack any knowledge of the Robber's location. Between each round of investigations, the Robber must move to an adjacent vertex. The minimum number of vertices that must be checked to guarantee the Robber's capture is called the Workday Number. We first bound the Workday Number using flow networks on the graph. Then, we find an optimal "golden" flow network that gives the best bound. Finally, we define a "desirable partition" of the vertices into three sets, A, B, and C, such that the subgraph on C has a 2-factor, the subgraph on A and B is bipartite and has matching number $|B|$, and there are no edges between A and C. The invariant Workday Number is then $2|B| + |C|$ for any desirable partition. Also, a unique desirable partition can be determined using golden flow networks. Putting these results together gives a method to determine the Workday Number for an arbitrary multigraph. Finally, we show that the Workday Number is a generalization of the matching number, by showing that it is the size of the largest subgraph which has a 2-factor.

1. Introduction

The classical mathematical problem of "Cops and Robbers on a graph" has been studied for decades. First introduced by Nowakowski and Winkler [5] and independently by Quilliot [6] Cops and Robbers is an area of deep mathematical research containing several hard open problems such as Meyniel's conjecture. Originally published by Frankl, Meyniel's conjecture states that the minimum number of Cops necessary to catch a Robber moving on a finite graph of n vertices is $O(\sqrt{n})$ [2].

Many different applications and mathematical insights have originated from analyzing variations of the original game. Different variations can account for imperfect information, varying speeds of the players, and movement restricted Cops which can then be applied as models of various important necessities ranging from firefighting to disease control (see [1]). In [7], Seymour demonstrates the usefulness of analyzing variations of Cops and Robbers by giving a game theoretical characterization of the treewidth, an important graph invariant in computer science. We analyze the case where the Cops have no information about the Robbers, and the Cops have "helicopters" so that they can move wherever they want.

In this variation of Cops and Robbers on a finite multigraph $G = (V, E)$ (inspired by [3]), the game is split into two steps, one where the Cops act (during the day) and one where the Robber acts (during the

Email address: ericds@princeton.edu (Eric Schneider)

Keywords: Cops and Robbers, graph invariant, Workday Number, desirable partition, golden flow network, 2-factor

night). First, the Robber is located on some vertex in V. The Robber's location is unknown to the Cops. Every night the Robber must move to an adjacent vertex. Every day, the Cops can check as many (or as few) vertices in G as they want to inspect and win if they find the Robber. The Workday Number, $W(G)$, is the minimum number of vertices that need to be checked by the Cops over all days in order to guarantee that the Robber is found. Note that the Workday Number is at most $|G|$ since the Cops could just check every vertex on the first day.

Throughout the paper, we will assume that the Robber is effectively omniscient so that the Robber will only be caught by the Cop's strategy if all of the Robber's strategies fail against the Cop's strategy. Additionally, we will use the term 'graph' to mean multigraph unless we explicitly refer to it as a simple graph.

In this paper, we completely determine the Workday Number for an arbitrary graph. We will first find a way using flow networks to bound the Workday Number. Next, we will define the optimal properties of a "golden" flow network, show how to compute one, and show that the bound provided by it is optimal. Then, we can define a desirable partition of the sets which has various properties. This will allow a new way of constructing graphs with known Workday Numbers. Afterwards, using golden flow networks, we will show that every graph can be constructed using these desirable partitions. Additionally, we will show how to calculate the invariant Workday Number using any desirable partition on the graph. Finally, we will show how the Workday Number is also a generalization of the matching number as the size of the largest subgraph containing a 2-factor.

2. Main Results

Definition 1. *A **flow network** on a graph is a weighted directed graph with the same vertices as the graph and with directed edges only between vertices if there was an edge originally between them.*

Definition 2. *A flow network is **special** if:*

1. *All directed edge weights are nonnegative real numbers.*
2. *The sum of the weights of the in-edges is equal to the sum of the weights of the out-edges for every vertex. Call this common sum the **weight** of the vertex.*
3. *All vertex weights are less than or equal to* 1.

Note that all directed edge weights of a special flow network are also less than or equal to 1 by the first and third conditions.

Definition 3. *For any special flow network F and subset A of the vertices of F, the expression $S(F, A)$ denotes the sum of the weights of all the vertices in A.*

Note that if A is V, the set of all vertices of G, then $S(F, V)$ is also the sum of the weights of the edges of F since every edge weight counts in the vertex weight sum for exactly one vertex.

Lemma 4. *For any special flow network F on a graph G, we have $W(G) \geq S(F, V)$.*

Proof. Let A_k be the set of vertices where the Robber could be if he was not yet found by the Cops after the k^{th} stage, so that A_{2i-1} is the set of vertices where the Robber could be after the i^{th} night, and A_{2i} is the set of vertices where the Robber could be after the i^{th} day. In particular, $A_1 = V$ before any Cop checks have been performed. The Robber is guaranteed to be caught once A_k is empty. Let C_k be the number of vertices checked by the Cops in total by the k^{th} stage, so that in particular $C_1 = 0$. Now in terms of A_k and C_k, $W(G)$ is the minimum value of C_k when A_k is empty over all possible Cop strategies.

Now, we will show that $S(F, A_k) + C_k$ is a nondecreasing function of k which when applied to the initial and final conditions will give us the desired result.

During the i^{th} day, the set of possible locations for the Robber decreases by at most the number of vertices checked, so

$$|A_{2i-1}| - |A_{2i}| \leq C_{2i} - C_{2i-1}$$

Since the weight of every vertex is at most 1 and $A_{2i} \subseteq A_{2i-1}$,

$$S(F, A_{2i-1}) - S(F, A_{2i}) \leq |A_{2i-1}| - |A_{2i}|$$

so

$$S(F, A_{2i-1}) - S(F, A_{2i}) \leq C_{2i} - C_{2i-1}$$

Thus, for any $i \geq 1$

$$S(F, A_{2i-1}) + C_{2i-1} \leq S(F, A_{2i}) + C_{2i}$$

as desired.

During the i^{th} night (for $i \geq 2$), A_{2i-2} is replaced by A_{2i-1}, where A_{2i-1} is the set of vertices adjacent to a vertex in A_{2i-2} since the Robber must move.

Let $w_{v,k}$ be a real number for every v in V and positive integer k such that if $v \notin A_k$ then $w_{v,k}$ is 0 and if $v \in A_k$ then $w_{v,k}$ is the weight of v in F.

Let $d_{v,k}$ be a real number for every v in V and every positive integer k such that $d_{v,k}$ is the sum of the directed edges coming into v from vertices in A_k.

By the definition of $S(F, A_k)$,

$$S(F, A_k) = \sum_{v \in V} w_{v,k}$$

Since $\sum_{v \in V} w_{v,k}$ and $\sum_{v \in V} d_{v,k}$ are both equal to the sum of the weights of all directed edges coming from vertices in A_k,

$$\sum_{v \in V} w_{v,k} = \sum_{v \in V} d_{v,k}$$

Now since $d_{v,2i-2}$ is nonzero only when v is adjacent to a vertex in A_{2i-2}, $d_{v,2i-2}$ is nonzero only if $v \in A_{2i-1}$. Therefore, if $v \notin A_{2i-1}$, $d_{v,2i-2} = 0 \leq w_{v,2i-1}$. Also, $d_{v,2i-2}$ is at most the sum of all edges coming into v, so $d_{v,2i-2}$ is at most the weight of v in F. Since for all $v \in A_{2i-1}$, $w_{v,2i-1}$ is the weight of v in F, if $v \in A_{2i-1}$, then $d_{v,2i-2} \leq w_{v,2i-1}$. Thus, for all $v \in V$,

$$d_{v,2i-2} \leq w_{v,2i-1}$$

Putting the equations together,

$$S(F, A_{2i-2}) = \sum_{v \in V} w_{v,2i-2} = \sum_{v \in V} d_{v,2i-2} \leq \sum_{v \in V} w_{v,2i-1} = S(F, A_{2i-1})$$

Since $C_{2i-2} = C_{2i-1}$, for all $i \geq 2$,

$$S(F, A_{2i-2}) + C_{2i-2} \leq S(F, A_{2i-2}) + C_{2i-1}$$

Thus, $S(F, A_k) + C_k$ is a nondecreasing function of k. Therefore, when the Cops perform optimally such

that $C_k = W(G)$ at the end,

$$\begin{aligned} W(G) &= S(F, \emptyset) + W(G) \\ &= S(F, A_k) + C_k \\ &\geq S(F, A_1) + C_1 \\ &= S(F, V) \end{aligned}$$

as desired. □

Corollary 5. *If there exists a special flow network F on a graph G, such that $|G| = S(F, V)$, then $W(G) = |G|$.*

The corollary follows immediately from Lemma 4 and the fact that $W(G) \leq |G|$. Lemma 4 and Corollary 5 can be utilized to solve several simple examples originally from [3].

Example. *The Workday Number of a graph with k disjoint edges is at least $2k$. In other words, the Workday Number is at least twice the matching number. To see this, let F be the special flow network where all directed edges between vertices in one of the k disjoint edges have weight 1. Then, since $S(F, V) = 2k$, $W(G) \geq 2k$ by Lemma 4.*

Throughout the paper we will adopt the convention that a cycle is a sequence of distinct vertices where any two consecutive vertices (including the first and last) are adjacent. This differs from the conventional definition only when a cycle has length 1 (a loop) or 2 (an edge).

Definition 6. *A graph has a **2-factor** if its vertices can be partitioned such that each part has a cycle containing all of the vertices of that part.*

Note that this definition follows the convention of [4] where degenerate 2-factors are still considered 2-factors.

Example. *The Workday Number of a graph G that has a 2-factor is $|G|$. To see this, let F be the flow network where each vertex in G has a directed edge of weight 1 to the next vertex in its cycle. Then, F is special and since $S(F, V)$=$|G|$, so $W(G) = |G|$ by Corollary 5.*

Example. *The Workday Number of a complete graph K_n is $|K_n| = n$. To see this, let F be the flow network where every directed edge has a weight of $1/(n-1)$. Then, F is special and since every vertex has weight 1, $S(F, V) = n$, so $W(K_n) = n$ by Corollary 5.*

We now turn our attention to using flow networks to determine the Workday Number for an arbitrary graph. First, we will show that a flow network can be computed that is optimal in several ways. Then, we will show that given this optimal flow network, there is a computable partition of the vertices of the graph satisfying certain properties. Lastly, we will show that if a graph is partitioned in this specific way, then the Workday Number can be determined. In addition to allowing the Workday Number to be determined for any graph, this provides a classification of all graphs and allows new graphs with known Workday Numbers to be constructed.

The first step is to define the properties of such an optimal flow network and show that one exists.

Definition 7. *A flow network is **symmetric** if whenever there is a directed edge from vertex a to vertex b with weight w, then there is also one from b to a with weight w.*

Definition 8. *A flow network F on a finite graph $G = (V, E)$ is **golden** if:*

1. *F is special and symmetric.*

2. *For all special flow networks F' on G, $S(F,V) \geq S(F',V)$.*
3. *For all special flow networks F' such that $S(F,V) = S(F',V)$, the number of vertices with weight 1 in F is less than or equal to the number of vertices with weight 1 in F'.*

Theorem 9. *For any graph, there exists a golden flow network. Additionally, for all golden flow networks, the set of vertices which have a weight of 1 is the same.*

Proof. Let us show that arbitrary graph G has a golden flow network. Let the set of edges of G be $E = e_1, ..., e_n$ where $n = |E|$. Let the set of vertices of G be $V = v_1, ..., v_m$ where $m = |V|$. Now, there exists a bijection between the space of all possible special symmetric flow networks on G to a subset of $\mathbb{R}^n$ where the vector $[w_1, ..., w_n]$ is matched with the symmetric flow network whose directed edges for e_k have weight w_k. For $1 \leq i \leq m$, let S_i be the set of integers k such that e_k is adjacent to v_i. By the definition of a symmetric special flow network, all such vectors must satisfy the following $m + n$ linear inequalities

$$0 \leq w_k \qquad \text{for } 1 \leq k \leq n$$

and

$$\sum_{k \in S_i} w_k \leq 1 \qquad \text{for } 1 \leq i \leq m$$

Thus, finding a special symmetric flow network that maximizes the sum of the weights of the graph is equivalent to an n variable linear programming question with $m+n$ constraints. Since there exists a solution to the linear programming question (let all n variables be 0) and the solution space is bounded (all weights are from 0 to 1 inclusive), there exists a vector in this space that maximizes the sum of the weights. Also, the set of all such solutions can be computed using linear programming.

Now, a vector in this set that minimizes the number of vertices of weight 1 must be chosen. First, the subset of the feasible region over which $S(F,V)$ is maximal is a convex combination of its corners. Now, consider the vertex weight of any vertex of the graph for the flow networks corresponding to the corners of the region. If a vertex has a weight of 1 in all of the corners, then that vertex has a weight of 1 throughout the entire region. However, if the vertex does not have a weight of 1 at any of the corners, then that vertex's weight is not 1 anywhere in the interior of the region (since all vertex weights are at most 1). Thus, all vertices which are not 1 somewhere are not 1 in a corner, so the interior of the region has the least number of vertices with weight 1. Thus, the flow network of every vertex in the interior of the maximal region is golden. All other golden flow networks, if there are any, must be on the boundary and have the same set of vertices which are not 1 as the golden networks on the interior. Since the special symmetric flow network associated with this vector has maximal sum of edge weights and minimal number of vertices of weight 1, it is a golden flow network. Thus, a golden flow network exists and can be computed. Also, if the minimal number of vertices have weight 1, then only those vertices which are 1 everywhere are 1 in a golden flow network, so the set of vertices which have a weight of 1 is the same for all golden flow networks. □

Now, we will describe properties of a partition that will make it desirable.

Definition 10. *A partition of the vertices of a graph into three sets, A, B, and C is **desirable** if:*

1. *There are no edges between elements of A and A or between elements of A and C.*
2. *There exists an injective function f from B to A such that for all b in B, $f(b)$ and b are adjacent.*
3. *C has a 2-factor.*

Graphs with desirable partitions are fairly easy to construct. First, consider several cycles with arbitrary edges between them. This will be set C. Then, consider a bipartite graph where every vertex on one side (B) is matched and adjacent to a unique vertex on the opposite side (A). Then, add arbitrary edges between B and A and between the vertices of B. Finally, arbitrary edges can be added between the vertices of B and C to give a graph with a golden partition determined by A, B, and C.

This process can be reversed to show that every graph with a desirable partition can be constructed in this way. First, all edges from C to C can be constructed using the first step by 3). Then, the bipartite subgraph between A and B can be constructed by first making the matching which is possible by 2) and then adding the rest of the edges. Finally, the remaining edges between vertices of B and C can be added, allowing the original graph to be constructed.

To illustrate this construction and several techniques that will be used throughout the rest of the paper, here is an example.

Example. *Consider the graph where C is a 3-clique, the subgraph induced by $A \cup B$ is $K_{2,3}$, and all of the vertices of B are connected with all of the vertices of C. In this graph, $|G| = 8$, $|A| = 3$, $|B| = 2$, and $|C| = 3$. Now let's calculate the Workday Number. Consider symmetric special flow network F with weights of 0.5 on all edges only between vertices of C and weights of 1 on edges between matched vertices of B and A. Then, $W(G) \geq S(F, V) = 2 \cdot 2 \cdot 1 + 6 \cdot 0.5 = 4 + 3 = 7$. $W(G)$ is 7 since 5 Cops could be placed on all of B and C the first day and 2 Cops on B the second day.*

In order to obtain a partition of the graph that is desirable using a golden flow network on that graph, we must first show several results relating properties of graphs to the properties of special flow networks.

Lemma 11. *For any bipartite graph $G = (U, V, E)$ and special symmetric flow network F on G such that the weight of every vertex in U is 1, there exists an injective function f from U to V such that for all $u \in U$, vertex $f(u)$ is adjacent to u.*

Proof. It will suffice to show that the bipartite graph satisfies Hall's Marriage condition.

For any subset U' of U, let V' be the subset of V consisting of vertices adjacent to a vertex in U'. Then, $S(F, U')$ is equal to the sum of the weights of the directed edges going to vertices in U' which is equal to the sum of the outgoing directed edges from vertices of U'. Since all such edges must go to vertices in V', then $S(F, U') \leq S(F, V')$. Using the fact that all vertices in U' have a weight of 1 and all vertices of V' have a weight of at most 1,

$$\begin{aligned} |U'| &= S(F, U') \\ &\leq S(F, V') \\ &\leq |V'| \end{aligned}$$

Thus, by Hall's Marriage Theorem such a function f exists. □

Corollary 12. *For any bipartite graph $G = (U, V, E)$ with $|U| = |V|$ and special symmetric flow network F on G such that the weight of every vertex in U is 1, there exists a bijective function f from U to V such that for all $u \in U$, vertex $f(u)$ is adjacent to u.*

Theorem 13. *A graph G has a special flow network F such that $S(F, V) = |G|$ if and only if G has a 2-factor.*

Proof. Assume that G has a special flow network F such that $S(F, V) = |G|$. Let H be a graph whose vertices are the disjoint union of two copies, V' and V'', of the set of vertices V. Let an edge go from a vertex g_1 in V' to a vertex g_2 in V'' when there is an edge between the vertices corresponding to g_1 and g_2 in G. Let J be a symmetric special flow network on H such that the weight of the directed edges between vertices g_1 and g_2 in V' and V'' respectively is the weight of the directed edge in F from the vertex corresponding to g_1 to the vertex corresponding to g_2.

Since all of the vertices in G have weight 1 from flow F, all vertices in H have weight 1 from flow J. Applying Corollary 12 gives that there exists a bijective function from V' to V'' where vertices are adjacent to their pair. Thus, there is a bijective function f from V to V such that for all $v \in V$, vertex $f(v)$ is

adjacent to v. Since f permutes the vertices of V and every permutation is a composition of disjoint cycles, it is possible to partition V into the vertices in each cycle and so V has a 2-factor. Note that this holds due to our convention for cycles. Even if f has fixed vertices, such a fixed vertex must be adjacent to itself (have a loop) and therefore, the subgraph induced by that vertex has a cycle containing every vertex in that part.

Now assume that G has a 2-factor. Let F be the symmetric special flow network that has weights of 0.5 on all directed edges between adjacent vertices in one of the cycles (or a weight of 1 for cycles of length 2). Then, every vertex has either two 0.5 contributions or one 1 contribution for its incoming edges, so the weight of every vertex is 1. Therefore, $S(F, V) = |G|$. □

Note: This theorem can also be shown using the Birkhoff-von Neumann Theorem since the adjacency matrix of F is a doubly stochastic matrix. Therefore, the adjacency matrix is the convex combination of permutation matrices. Choosing one of these permutation matrices used in the sum gives that there is a permutation of the vertices of G such that every vertex of G is adjacent to the vertex it goes to. The proof then proceeds as above.

Finally, let's define a way of categorizing vertices that will be useful when creating desirable partitions from golden networks.

Definition 14. *In a special flow network, a vertex is* ***hard*** *if it has a weight of* 1 *and is only adjacent to vertices with weight* 1*.*

Definition 15. *In a special flow network, a vertex is* ***soft*** *if it has a weight of* 1 *and is adjacent to a vertex with a weight less than* 1*.*

Note that every vertex is either soft, hard, or has a weight not equal to 1. Now, we will define how desirable partitions can be related to golden flow networks and then show some results relating golden flow networks and desirable partitions.

Definition 16. *A partition of the vertices of a graph into three sets, A, B, and C, is* ***derived*** *from a special flow network if the sets A, B, and C of the partition are the vertices with weight not equal to 1, soft vertices, and hard vertices respectively of the flow network.*

Lemma 17. *For any graph, there is a unique partition of the vertices of that graph into three sets that is derived from all golden flow networks.*

Proof. By Theorem 9, all golden flow networks have the same set of vertices that have weight 1. Then, the set of vertices that don't have weight 1 is the same for all flow networks. Therefore, the set of soft vertices and hard vertices is the same for all golden flow networks. Thus, the partition derived by all such golden networks is unique and has the set of all vertices that don't have weight 1 in A, all of the soft vertices in B, and all of the hard vertices in C. □

Now, we will show that given a golden flow network on a graph, we can compute a desirable partition on that graph.

Theorem 18. *The partition derived from a golden flow network is desirable.*

Proof. Consider a golden flow network F on graph G. Let A be the set of all vertices with weight in F not equal to 1. Let B be the set of all soft vertices of F. Finally, let C be the set of all hard vertices of F. We will show that this partition is desirable by showing it satisfies each of the three necessary properties.

First, we will show 1). By definition of hard vertices, there are no edges from A to C. Also, there are no edges between two vertices in A with weight not equal to 1 since if there were an edge between two vertices with weights w_1 and w_2, the weight of the edge could increase by 1-max(w_1,w_2). This would increase the total sum which would contradict the maximality of the sum.

Now, we will proceed to show 2). First we will show that in F, all edges from B to B or B to C have weight 0. Assume for the sake of contradiction that there are adjacent vertices b in B and c in B or C such that the directed edges between them have a nonzero weight. Then, since b is soft, there is a vertex a in A with weight w_a which is adjacent to b. Let a and b be connected by edges with weight w_1, and b and c be connected by edges with weight w_2 where w_2 is nonzero. Then, we can construct a flow network F' where the weights of all edges not between a, b, and c are the same. Let ϵ be $min(w_2, 1 - w_a)/2$. The edges between a and b in F' have weight $w_1+\epsilon$ and the edges between b and c in F' have weight $w_2 - \epsilon$. Since $S(F, V) = S(F', V)$, and the number of vertices with weight 1 is lower in F' than in F, the minimality of the number of 1's in F is contradicted. Thus, all edges from B to C have weight 0.

This means that all weighted edges connected to B are to A and not to B or C. Therefore, let $G_{A\cup B}$ be the subgraph of G induced by $A \cup B$ and $F_{A\cup B}$ be the subgraph of F induced by $A \cup B$. Every vertex of B has weight one in F and all nonzero edges of F are in $F_{A\cup B}$. Therefore by Lemma 11, there is an injective function f from B to A such that for all $b \in B$, we have $f(b)$ is adjacent to b.

Finally, let us show 3). Since there are no nonzero edges from C to B or C to A, all of the weights of the vertices of C comes from edges from vertices in C to vertices in C. Also, since all of the vertices of C are hard, these weights are all 1. Let G_C and F_C be the subgraphs of G and F induced by C respectively. All vertices of C have weight 1 in F and so in F_C. Since there are no edges of nonzero weight from C to B or C to A, we must have that $S(F, C) = |C|$. Thus, by Theorem 13 we know that C has a 2-factor. □

Now we can combine the results of Theorems 9 and 18.

Corollary 19. *Every graph has a computable desirable partition.*

Theorem 20. *For any desirable partition of the vertices of G into A, B, and C, the Workday Number of G is $|C| + 2|B|$.*

Proof. Let flow network F be a symmetric flow network on G such that there is an edge of weight 1 from b to $f(b)$ for all $b \in B$. Also, let the weights of the edges of C be those determined by the flow network on the subgraph of G induced by C from Theorem 13.

By Theorem 13, we have $S(F, C) = |C|$. Since there are $2|B|$ edges of weight 1 from A to B (two for each vertex of B), $S(F, A \cup B) = 2|B|$. Thus, $S(F, G) = S(F, A \cup B) + S(F, C) = 2|B| + |C|$.

Applying Lemma 4 gives that the Workday Number of the graph is at least $2|B| + |C|$.

Now, let us show that $2|B| + |C|$ is attainable. On the first day, put Cops on all of B and C, and on the second day, put Cops on all of B. This takes $2|B| + |C|$ Cops. If the Robber was on B or C, then he would be caught on the first day. If he was on A, then he must be on B on the second day, and hence would be caught. Thus, the Workday Number is $2|B| + |C|$ as desired. □

This shows that the Robber can be caught within 2 days using the Workday Number of Cops. The only time in which it is possible to catch the Robber in one day using the minimum Workday Number of Cops is when the Workday Number is equal to the size of the graph ($|A| = |B|$).

Lastly, we will show that the Workday Number is a generalization of the matching number. Since a perfect matching is also known as a 1-factor, the matching number is half of the size of the largest subgraph that has a 1-factor.

Theorem 21. *The Workday Number is the size of the largest subgraph that has a 2-factor.*

Proof. First, by Corollary 19, every graph has a desirable partition. Therefore, consider the subgraph with vertices from C, B, and the image of the injective function, f, from B to A. This subgraph has size $2|B|+|C|$ which is the Workday Number. Also, this subgraph has a 2-factor. First, since C has a 2-factor, its vertices can be partitioned such that each part has a cycle containing all of the vertices of that part. Then, the rest

of the subgraph can be partitioned into pairs $\{b, f(b)\}$ for all b in B. Thus, the size of the largest subgraph with a 2-factor is at least the Workday Number.

Consider any subgraph H of G that has a 2-factor. By Theorem 13, there is a special flow network F on this subgraph such that the sum of its weights is equal to the subgraph's size. Let F' be the extension of this flow network to the entire graph by making the weights of the rest of the edges 0. By Lemmas 4 and 11, $|H| = S(F, H) = S(F, G) \leq W(G)$, so the size of a subgraph with a 2-factor is at most the Workday Number of the graph.

Combining these results shows that the Workday Number is equal to the size of the largest subgraph that has a 2-factor. □

Theorem 21 shows how this variant of Cops and Robbers is related to a graph invariant. This relationship is similar to how the Cop number in a different variation of Cops and Robbers is equal to the treewidth [7].

3. Graph Theoretic Approach

While the connection between flow networks and the Workday Number is interesting and may have further applications, Theorem 21 is independent of the machinery of linear programming or flow networks. Therefore, it would be interesting if there was an approach using purely Graph Theoretic methods to show Theorem 21. We answer this affirmatively with the below proofs.

Lemma 22. *If the Robber can be caught within n days with less than or equal to k checks, then the Robber can be caught within 2 days using at most k checks.*

Proof. We will proceed by Induction on n:

For the base case, if n is 1 or 2, then the Robber can be caught within 2 days with the same number of checks as desired.

Now, we shall proceed with the inductive step. Now consider a set of at most k checks that captures the Robber in n days. Let all of the checks performed on day n be performed on day $n-2$. Let us show that this new strategy does in fact catch the Robber.

Assume for the sake of contradiction that there are vertices v_1 and v_2 such that v_1 and v_2 are adjacent and the Robber would not be caught with this new strategy if he were on v_1 on day $n-2$ and on v_2 on the day $n-1$ day. Then, a Robber could go from v_1 to v_2 to v_1 and not get caught with the original strategy giving us a contradiction. Thus, the new strategy that takes $n-1$ days does in fact catch the Robber. Applying the inductive hypothesis, since there is now a strategy with less than or equal to k checks that catches the Robber in $n-1$ days, there is one that catches the Robber with k checks in 2 days as desired. □

Now we will show another proof of Theorem 21.

Theorem 21 . *The Workday Number is the size of the largest subgraph that has a 2-factor.*

Proof. The Workday Number is equal to the smallest number of cop checks necessary to catch the Robber. By Lemma 4, the Workday Number is the smallest number of cop checks necessary to catch the Robber within 2 days.

Let H be the bipartite graph whose vertices are the disjoint union of two copies, V' and V'', of the set of vertices V of G. Let an edge go from a vertex g_1 in V' to a vertex g_2 in V'' when there is an edge between the vertices corresponding to g_1 and g_2 in G. Then, the cops check on 2 days. The check on the first day prevents the Robber from being on a subset of the vertices in V' while the second day's check prevents the Robber from being on a subset of V''. Thus, the Robber can escape detection if and only if there is an edge of H of which neither vertex was checked. Thus, the cop checks correspond to a vertex cover of H. Therefore, the Workday Number is the size of the smallest vertex cover of the bipartite graph. By König's

Theorem, the Workday Number, which is the size of the minimum vertex cover of H, is equal to the size of the maximum matching of H.

Now we will show that the size of the maximum matching on H is equal to the size of the largest subgraph of G that has a 2-factor.

First, we will show that there is a subgraph with a 2-factor which has a size at least the Workday Number. Consider the subgraph of G whose edges are determined by the matching. For every pair $g_1 \in V'$ and $g_2 \in V''$ in the matching, let the vertices corresponding to g_1 and g_2 in G be connected by an edge. Since every vertex has degree 0, 1, or 2, this partitions the graph into isolated vertices, paths, and cycles. Since the isolated vertices are unrelated to the edges of the maximum matching, we remove the isolated vertices from the subgraph. By counting where the edges of the matching go, a cycle of size n corresponds to n edges of the matching while a path of size n corresponds to $n-1$ edges.

If there were a path with an $2n$ vertices, then it would use $2n-1$ edges of the matching. However, by pairing up all of the adjacent vertices of the path, breaking the $2n-1$ edges of the matching, and creating n sets of 2 edges between the adjacent vertices, the matching number could be increased which contradicts the maximality of the matching.

Thus, there are only paths with an odd number of vertices and cycles. For paths with an odd number, $2n+1$, of vertices, partition the vertices again into n pairs each of which is a cycle by our convention. The total number of vertices that are incorporated into the subgraph is $2n$ which is $2n+1-1$, so that the number of vertices used is equal to the number of edges of the matching that were originally associated with the path. Therefore, we can break up this subgraph into cycles such that the number of vertices in all of the cycles is equal to the number of edges in the maximum matching. Thus, there is a subgraph with a 2-factor which has a size at least the Workday Number.

Now, for any subgraph of G that is a 2-factor, we can give all of the cycles an orientation. Then, let f be the bijective function from vertices in the subgraph to vertices in the subgraph such that for any vertex g in the subgraph, $f(g)$ denotes the next vertex after g in the cycle. Then, for all g in the subgraph just connect the vertex corresponding to g in V' with the vertex corresponding to $f(g)$ in V''. Since f is bijective and g and $f(g)$ are adjacent, this is in fact a matching. Therefore, the Workday Number is at least the size of any subgraph with a 2-factor.

Thus, the size of the largest subgraph of G containing a 2-factor is $W(G)$. □

Note that since there exist an algorithm that can find the maximum matching of a bipartite graph in $O(\sqrt{|V|}|E|)$, by applying that algorithm with the above proof, we can find the Workday number in $O(\sqrt{|V|}|E|)$.

4. Conclusion

In summary, we have shown how to compute the Workday Number for arbitrary graphs. First using linear programming, a golden flow network on the graph can be computed. Next, a unique desirable partition can be derived from any golden flow network. The concept of a desirable partition is purely a property of the graph independent of the game and so may be useful in further research into this and other variants of Cops and Robbers. Additionally, the concept of special flow networks may be useful in future research into Cops and Robbers variations with incomplete knowledge since the flow network is able to encode the possible locations and movements of the Robber. Lastly, the determination of a generalization of the matching number through these results parallels the game theoretic characterization of treewidth. Hopefully, future research will show how to find and better understand the next generalization of the matching number, the size of the largest subgraph having a k-factor.

5. Acknowledgements

I would like to thank Ken Monks for all of his guidance, advice, and support. I would also like to thank my parents for all of their support.

References

[1] A. Bonato and R. Nowakowski, *The Game of Cops and Robbers on Graphs*, American Mathematical Society, Providence, Rhode Island, 2011.

[2] P. Frankl, "Cops and Robbers in graphs with large girth and Cayley graphs", *Discrete Appl. Math.* 17 (1987), no. 3, 301-305.

[3] Karafiol, Paul J., Paul Dreyer, Edward Early, Zuming Feng, Benji Fisher, Zachary Franco, Chris Jeuell, Winston Luo, Andy Niedermaier, Andy Soffer, and Eric Wepsic. "ARML Competition 2012." *The Official American Regions Mathematics League Web Site*. Last modified June 1, 2012. http://www.arml.com/2012_contest/2012_Contest_Final_Version.pdf.

[4] Meijer, Henk, Yurai Nez-Rodrguez, and David Rappaport. "An algorithm for computing simple k-factors." *Information Processing Letters* 109, no. 12 (May 31, 2009): 620-25.

[5] R.J. Nowakowski, P. Winkler, "Vertex-to-vertex pursuit in a graph", *Discrete Mathematics* 43 (1983), 235-239.

[6] A. Quilliot, "Jeux et pointes fixes sur les graphes", Ph.D. thesis, Université de Paris VI, 1978.

[7] P. D. Seymour and Robin Thomas, "Graph searching and a min-max theorem for tree-width"', *J. Combin. Theory Ser. B* 58 (1993), no. 1, 22-33.