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

Eric Schneider
High Technology High School
Freehold, NJ

July 1, 2013

Personal

Oftentimes, there are "bad guys" such as diseases, wildfires, or thieves that the "good guys" such as the CDC, firemen, or police wish to control or capture. However, the good guys only have a limited quantity of assets such as money, people, and time, so it is important for them to use the least amount of resources. One well-known way of analyzing such problems is known as "Cops and Robbers on a Graph". I analyzed a different version of this model to find out how to minimize the cost (called the Workday Number) to catch the bad guys. I discovered how to compute a way to catch the bad guys in two days while still minimizing the cost.

How did I come up with my research topic? During the proof-based power round of the national American Regions Mathematics League Competition (ARML), there was one problem which introduced and asked questions about the Workday Number. In answering it, I realized that I could combine the idea of flow networks from computer science with monovariants from my math experience to give bounds on the Workday number. Unknown to me at the time, the panel of judges, all mathematicians, had deliberated for over thirty minutes over the correctness of my solution. Although it appeared correct, and they could not find any holes in it, it simply did not match any of the official proofs that they had. I received full credit but my coach and mentor, Professor Ken Monks, seemed intrigued by what I had done and asked me if I could explain and expand upon my solution. In doing so, I decided to generalize my technique in order to increase my own understanding of the problem. As I conducted further research and spent countless hours thinking about it, I noticed several interesting connections between this problem and the rest of graph theory. My research into graph theory heightened my awareness of how different areas of math and science are related. Even something as simple as a variant of the Pac-man video game has applications in computer science, mathematics, mine searching, and cleaning systems. As my understanding of various aspects of such different disciplines as computer science and mathematics has evolved, the overlap between them in graph theory and the similarities between them have shown how problems that may arise in one discipline can often be solved using techniques and ideas from other fields.

While it started out as an innovative idea strictly in pure mathematics, as I developed it, I continued to have new insights into the solution and its potential practical applications. For example, cities plagued by uncontrolled wildfires which can endanger vegetation, lives, and homes urgently need more efficient solutions. Stopping the spread of disease is another issue. Thus, while completing my research, I was inspired by the possibility of creatively using sophisticated math techniques to discover solutions to current issues.

Background: Graph Theory

Graph theory is a branch of mathematics that is extremely useful in modelling communication networks, data organization, linguistics, and computational flow. The basic idea is that there are many points (also known as vertices or nodes) and edges between these locations. For example, the points could be buildings while the edges could be roads between them. Another possible scenario that could be modeled using graphs is when the points are trees and the edges represent trees close enough so that one could catch fire from another. In my research, a graph represents the set of hiding places for the bad guys.

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 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 $W(G) \geq S(F, V)$.*

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.*

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.*

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.*

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.*

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.*

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 $|C| + 2|B|$.*

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.*

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. In fact, the below theorems can be proven using only graph theory.

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.*

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

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 theorem, 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. Finally, this research may lead to insights into real life situations with incomplete information such as stopping the spread of disease and capturing hidden criminals.

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.