Maximum Matching Information Delivery in Ad-hoc Networks

Mesut Ozdag, Madhura Joshi
College of Engineering and Computer Science
University of Central Florida
{mesut.ozdag, madhurajoshi} @knights.ucf.edu

***Abstract* – It has been very popular to do research on communication services due to the demand for interaction with anyone, anytime and anywhere. There are a lot of problems with communication systems which graph theory address. Information delivery problem takes one of the main roles to provide the best communication between devices in various networks. With this paper, we present a graph theoretical approach for an information delivering problem in ad-hoc networks. There has been a couple of algorithms developed to find minimum-weighted tree out of a given graph, such as Prim's algorithm and Kruskal's algorithm. In addition, maximum matching problem has been solved by various methods, such as Edmonds algorithm and Kuhn's algorithm. In this paper, we demonstrate a synthesis of Prim's algorithm and Edmonds algorithm to obtain the best solution for information delivery problem.**

I. Introduction

Recent work shows that ad-hoc networks have become very significant and influential in several applications on information delivery problem. An ad-hoc network is a temporary connection between computers and/or devices used for a specific purpose, such as playing multiplayer computer games and sharing documents during a meeting. It supports anytime and anywhere computing. It is a collection of two or more devices composed of networking capability and wireless communications. Wi-Fi Internet Network and mobile networks are considered as ad-hoc networks.

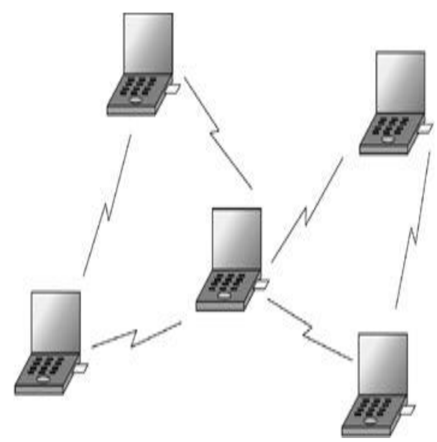


Figure.1: An example of an ad-hoc network.

Mobile communication has been one of the most active research in communications in several decades. Information delivery problem occurs when we form a connection from a vertex to another one. In order to build a network, we need the minimum number of such connections that has no vertex left with no information. Thus, we obtain the minimum cost for information delivery through the network. For example, coloring algorithms are applied to the channel assignment problems [1]. In this paper, we focus on maximum matching algorithm by Edmonds for information delivery problem [2].

Information delivery problem can be addressed in three ways [3]:

1. ***Spanning tree:*** We aim on delivering a set of information to the all nodes in a graph such that there is always a way for two nodes to connect.

2. ***Maximum cardinality matching for general graphs:*** Our goal is to connect each node to some other node in the graph such that no node is left disconnected [4, 5, 6, 9].
3. ***Maximum bipartite matching:*** If there are two sets of information hubs, then connecting nodes in such a way that each node in one set is connected to all nodes in second set [7, 8].

We propose our solution as follows: Firstly, we aim on finding minimum connections needed to deliver information between vertices. We solve this problem with minimizing the cost of delivering the information which is actually finding the Minimum Spanning Tree (MST). We apply *Prim's algorithm* for MST. Next, we find a match for each node in the network. Lastly, we maximize the matches by *Edmonds algorithm* so that no vertex is left unmatched.

The rest of the paper is organized as follows: In Section II, we introduce the prior works in maximum matching algorithm in information delivery problem. Section III presents our solution for information delivery problem with a combination of Prim's algorithm and Edmonds algorithm. Lastly, Section IV concludes the paper.

II. Related Work

We find matches by forming sets of vertex disjoint edges in a graph. It has been a significant research topic for over fifty years. Particularly, it has been important to find maximum matching which means the matching of maximal cardinality. Its applications are multidisciplinary; not only in computer science, but also in economics, biology, and chemistry.

The first works on matching were based on un-weighted bipartite graphs (matching men and women). Perfect matching was presented by Hall's marriage theorem. On the other hand, weighted bipartite matching problem was firstly presented by Kuhn's algorithm (the Hungarian method, 1955). The algorithm works particularly by attempting to construct the current matching M to find a larger one by finding an M-augmenting path. Kuhn's main theorem demonstrates that a graph matching is maximum if there is no augmenting path in the graph. This algorithm is very fast for bipartite graphs. However, the algorithm leads problems to find the present augmenting paths on non-bipartite graphs. Conversely, Edmonds used augmenting paths to find a maximum matching in his work [2]. He was the first person who proved that a maximum matching can be reached in polynomial time.

Maximum matching in bipartite graphs is much simpler than in general graphs because computations of augmenting paths do not run into odd cycles.

Much work on matching has been done over years. Some works use the idea of Hopcroft and Karp's algorithm which uses Local Computation Algorithms (LCA) in order to receive an edge-query to determine if that edge is a part of the matching [7]. Ranking algorithm is also applied on random graphs for a near-perfect matching [4]. In addition, an $O(rn^2 logn)$-time algorithm is presented to find a maximum matching in r-regular graphs with ne vertices [5]. Furthermore, a fully dynamic algorithm approach is demonstrated by processing edge insertion and deletion, but it is still challenging to maintain the time complexity for c-approximate matching with $c<2$ [6]. Since the performance of multithreaded computers are sensitive to the order of vertex matching, augmentation-based parallel matching algorithms are also developed [8].

There has been some works using Edmonds algorithm which comes up with more advantages, such as providing a mesh which assures to be quadrilateral, giving optimal solution in a certain way, and faster performance [9].

III. Towards The Synthesis of Solutions

As we stated in Section I, we take our initial step with finding MST with Prim's algorithm. Next, the method we present here is based on the famous algorithm proposed by Edmonds [2] which allows to find minimal cost perfect matching of a given graph.

A. *Finding Minimum Spanning Tree with Prim's Algorithm*

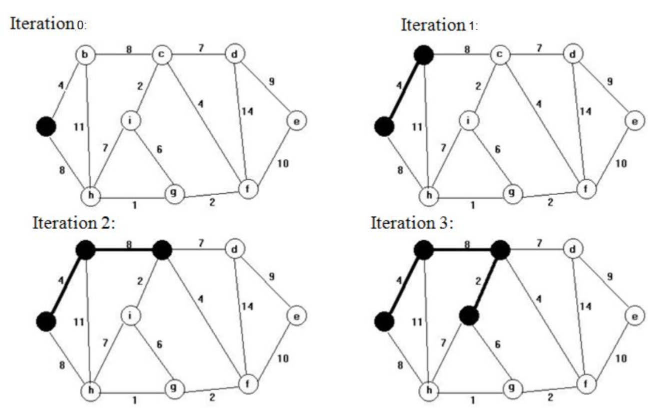
Prim's algorithm is a greedy algorithm that finds an MST for a weighted graph. In other words, it gives us a set of edges that is a tree with the minimum total weight which has every vertex.

Algorithm.1*: Prim-MST (G)*
Input*: Graph G=(V,E) with edge-weights.*

1. *Initialize MST to vertex 0.*
2. *priority[0] = 0*
3. *For all other vertices, set priority[i] = infinity*
4. *Initialize prioritySet to all vertices;*
5. ***while** prioritySet.notEmpty()*
6. *v = remove minimal-priority vertex from prioritySet;*
7. ***for each** neighbor u of v*
// Explore the edge.
8. *w = weight of edge (v, u)*
9. ***if** w < priority[u]*
10. *priority[u] = w*
11. ***endif***
12. ***endfor***
13. ***endwhile***

Output*: A minimum spanning tree of the graph G.*

Algorithm 1 shows how to find MST by Prim's algorithm. Initially, we select a random vertex from a given graph. Next, we select the shortest edge connected to that vertex. Then we choose the shortest edge connected to any vertex which is already connected under the condition that we do not form any cycle. We repeat this step until all vertices has been connected. That gives us minimum spanning tree with the smallest number of weight (Figure.1).



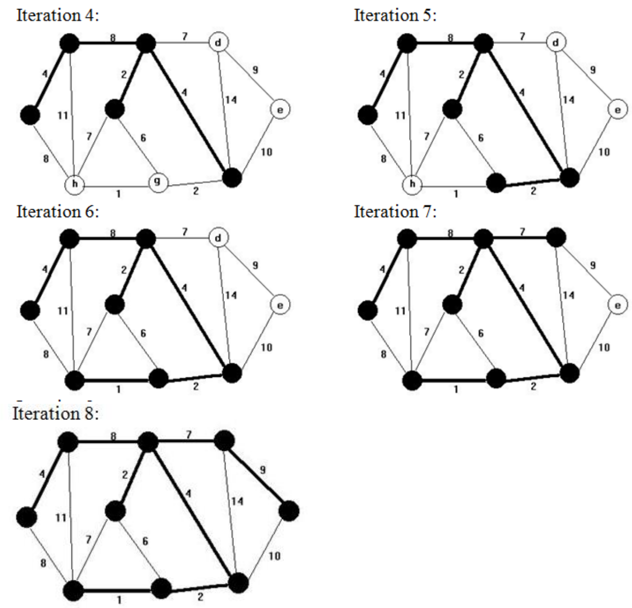


Figure.1: An example of finding Minimum Spannig Tree by using Prim's algorithm. Here, the iteration 8 represents the optimal solution of the given graph. The minimum length is 4+8+7+9+2+4+2+1=37.

B. *Maximum Matching using Edmonds Algorithm*

Given a graph G with vertex set V and edge set E. A matching is defined as a set of disjoint edges M in G. In other words, no two edges share a common vertex. Moreover, we define maximum matching M' as a matching for which M has the maximum number of set of non-touching edges found for a particular G.

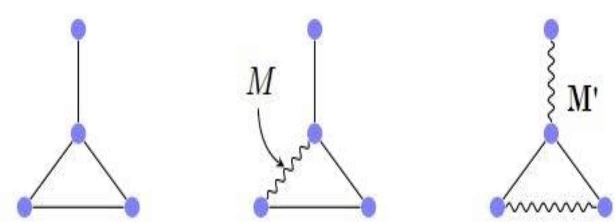


Figure.2: An example of maximum matching M'.

Terminology:

Flower: A flower is formed by two main sub-elements: a blossom and a stem.

Blossom: It is defined as an odd length cycle which gives an alternative matching. The base of the blossom is basically the only vertex that is not covered by the matching.

Stem: It represents an even length alternating path on the current matching.

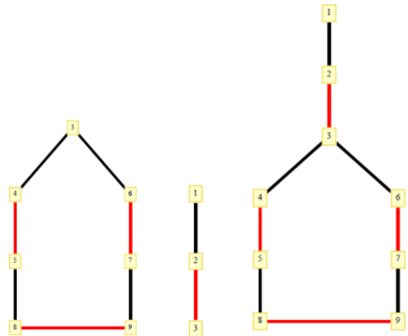


Figure.3: An example of a blossom, a stem, and a forest respectively. Vertex number 3 is the base of the blossom.

Alternating Path: An alternating path in (G, M) is a simple path whose edges are alternately in M and not in M.

***Augmenting Path**:* An augmenting path is an alternating path whose ends are distinct exposed vertices.

***Algorithm.2**: Edmonds Algorithm*

1. ***for** all $v \in V$, v is exposed*
2. *Search for simple alternating paths starting at*
3. *v*
4. *Shrink any found blossoms*
5. ***if** path P ends at an exposed vertex*
6. *P is an augmenting path*
7. ***{Update M}***
8. ***else if** no augmenting paths found*
9. *Ignore v in future searches*
10. ***end if***
11. ***end for***

Current M is maximum {No more augmenting paths}

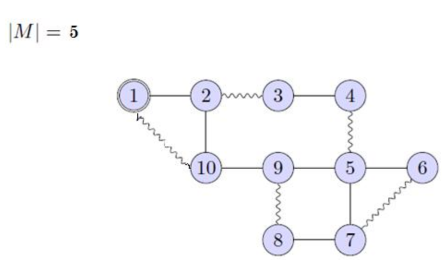


Figure.4: An example of maximum matching by using Edmonds algorithm. (Complexity: $O(n^4)$.)

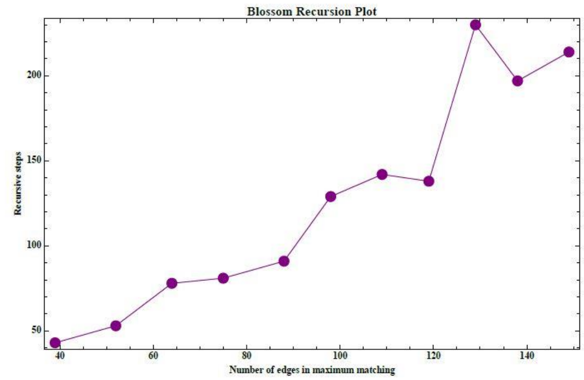


Figure.5: Blossom recursion plot.

```
Enter the number of vertices : 4
Enter number of edges : 5
Enter edges in u, v, w format where the edge is between vertices u, v with weigh
t w
Enter the edge details : 0 1 4
Enter the edge details : 0 2 7
Enter the edge details : 1 3 8
Enter the edge details : 2 3 1
Enter the edge details : 3 0 2
Prim's algorithm
Enter the starting vertex : 1


Printing MST
Edge Weight
1 0 4
3 2 1
0 3 2


Edmond's algorithm
Edges
{0: 1, 1: 0, 2: 3, 3: 2}
```

Figure.6: An example of the output we obtain.

```
Enter the number of vertices : 6
Enter number of edges : 7
Enter edges in u, v, w format where the edge is between vertices u, v with weigh
t w
Enter the edge details : 0 1 3
Enter the edge details : 0 2 4
Enter the edge details : 1 2 9
Enter the edge details : 2 3 2
Enter the edge details : 2 4 5
Enter the edge details : 3 4 7
Enter the edge details : 4 5 1
Prim's algorithm
Enter the starting vertex : 1


Printing MST
Edge Weight
1 0 3
0 2 4
2 3 2
2 4 5
4 5 1


Edmond's algorithm
Edges
{0: 1, 1: 0, 2: 3, 3: 2, 4: 5, 5: 4}
```

Figure.7: An example of the output we obtain.

IV. RESULT

Mobile communication networks have been very popular research area in recent years. This paper presents the design, implementation, and the evaluation of a synthesized solution to obtain better solution for information delivery problem on ad-hoc networks.

There has been many solutions for maximum matching on various types of graphs, such as random graphs, regular graphs, and mostly on general graphs. The most interesting open problem is whether a faster algorithm exists or not. In this case, parallel algorithms are going to be in demand for better performance in order to solve significant problems on maximum matching in networks and graph theory.

V. REFERENCES

[1] Sengoku M., Tamura H., Shinoda S., Abe T., "Development in Graph and/or Network Theoretic Research of Cellular Mobile Communication Channel Assignment Problems," IEICE Trans. Fund., 1994.

[2] Edmonds J. Maximum matching and a polyhedron with 0-1 vertices. J. of Research at the National Bureau of Standards, 1965.

[3] M. Sengoku, H. Tamura, K. Mase, S. Shinoda, "A routing problem on ad-hoc networks and graph theory," IEEE, 2000.

[4] M. Poloczek, M. Szegedy, "Randomized Greedy Algorithms for the Maximum Matching Problem with New Analysis," 2013.

[5] R. Yuster Maximum matching in regular and almost regular graphs, Springer-Verlag, 2012.

[6] S. Baswana, M. Gupta, S. Sen, Fully dynamic maximal matching in O(logn) update time, 2011.

[7] Y. Mansour, S. Vardi, "A local computation approximation scheme to maximum matching," 2013.

[8] A. Azad, M. Halappanavar, S. Rajamanickam, E. Boman, A. Khan, A. Pothen, "Multithreaded algorihms, for maximum matching in bipartite graphs," IEEE, 2012

[9] J. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, C. Geuzaine, "Blossom-Quad: a non-uniform quadrilateral mesh generator using a minimum cost perfect matching algorithm," 2012.