BOĞAZİÇİ UNIVERSITY

INDUSTRIAL ENGINEERING DEPARTMENT

IE 515 – GRAPHS AND NETWORK FLOWS

FALL ‘18

PROJECT REPORT

ALİM BUĞRA ÇINAR

2018702015

Contents

1. Introduction

This project is created as a part of IE 515 – Graphs and Network Flows class of Boğaziçi University Industrial Engineering department. The aim of the project is implementing network flow algorithms and compare their performances. To achieve that aim, a minimum cost flow problem is modeled and solved by using two basic minimum cost flow algorithms namely cycle-cancelling and successive shortest path algortihms. The model used as an example problem is building evacuation model which is a dynamic flow application.

In dynamic flows, the underlying solution idea is basically modeling the network as a static network, and creating its time expanded replicas. In building evacuation case, a building is represented as a static network. The nodes represent locations of buildings such as offices, work centers, hallways etc. The arcs represent the passages between those locations. Arcs have capacity and cost attributes. Capacities and costs respectively represent the number of people that can pass per unit time, and time required for passing that arc.

Time expanded replica is created by simply creating a copy of the static network for each time period. Then, in the replicated network, each node will have a copy of number of time periods which are represented by (i,p). In the replica, arcs are connected to the nodes according to their costs. For example, let the cost of arc (i,j) in the static network be t. Then, for every (i,p) node in the replicated network, there will be an arc ((i,p),(j,p+t)).

The aim of the model is finding the minimum time required for the building to be evacuated. We can find that objective by solving a minimum cost flow problem in the replicated network. We can model the minimum cost flow problem by adding super-source and super-sink nodes. Super-source will be connected to each replica of source with uncapacitated arcs of zero cost. Additionally, super-sink will be connected nodes with uncapacitated arcs of increasing cost. Supply of super-source and demand of super-sink will represent the number of people to be evacuated. In the solution of the minimum cost flow problem on the modified replicated model, the sink node with the maximum time period connected to super-sink will give us the minimum time needed to evacuate the building.

2. Approach

As it is mentioned in the introduction, two minimum cost flow algorithms will be implemented to solve the building evacuation model. The first algorithm implemented in cycle-cancelling algorithm. Cycle-cancelling algorithm firstly establishes a feasible flow in the network. To find the feasible flow, a max-flow problem on the network is solved. In the max flow problem, a source node connected to supply nodes with arcs of capacity equal to the supply and cost zero. Also, a sink node is connected from each demand node with arc capacities equal to their demand with cost of zero. Then, by solving a max flow problem on the modified network, the feasible flow is found. To find the feasible flow, max-flow labeling algorithm is implemented. After establishing feasible flow, negative cycles are found and flows augmented on negative cycles until no negative cycles left. To find negative cycles, Bellman-Ford algorithm is implemented.

The second algorithm implemented is successive shortest path algorithm. In this algorithm, the idea is finding shortest paths on the network with respect to reduced arc costs, and augmenting flow on the shortest paths until all demand is satisfied. The algorithm solves a shortest path problem in each iteration, and shortest paths are found by implementing Bellman-Ford algorithm.

3. Analysis of the Algorithms

We know from the negative cycle optimality conditions that if there is no negative cycle, then the network is optimal. Cycle-cancelling algorithm strictly finds a negative cost cycle and augments flow on the cycle until no negative cycles left. Therefore, the algorithm converges to a optimal solution. The number of iterations required for the algorithm in the worst case is 2mCU where m, C and U are number of arcs, max cost, and max capacity in the network. At each iteration, we find negative cycles by solving a shortest path problem by using Bellman-Ford algorithm, which is of complexity O(nm). As a result, overall complexity of the algorithm is O(nm^{2}CU) which is a pseudopolynamial complexity.

The successive shortest path algorithm sends flow from excess nodes to deficit nodes at each iteration. Therefore, maximum number of iterations required is nB where n is the number of nodes and B is the maximum supply in the network. At each iteration we solve a shortest path problem. Thus, the complexity of the algorithm is O(nB*S(n,m,C)) where S(n,m,C) represents the complexity of the shortest path algorithm used. In this project, the algorithm used is Bellman-Ford, therefore, successive shortest path algorithm has the complexity of $O(n^2mB)$. From the theoretical analysis, we expect successive shortest path algorithm to be faster than cycle-cancelling algorithm.

4. Implementation, Results & Discussion

OS: Windows 10 Home Version 1809 Build 17763 64 Bit

Processor: Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz, 1900 Mhz

RAM: 12 GB

Environment: Microsoft Visual Studio Community 2017 Version 15.9.5

All the implementation and test work is done on the system configured as shown above. The implementation of all codes are done by me except the structure of linked-list implementation. The general structure of linked-list and createnode(), display(), insert_start() and delete_first() functions are taken from the github repository of Kamal Choudhary[1]. Rest of the codes are created by my work.

[1] https://github.com/kamal-choudhary/singly-linked-list

Problem No	n	m	Successive Shortest Path	Cycle Cancelling	Ratio	Max B
1	4	5	0.06	0.43	7.166667	
2	6	10	0.25	0.56	2.24	
3	47	54	1.37	3.31	2.416058	
4	102	152	3.49	23.34	6.687679	50
5	102	152	5.36	31.53	5.882463	100
6	102	152	7.65	31.96	4.177778	150
7	102	152	9.22	45.13	4.894794	200
8	102	152	10.58	43.75	4.135161	250
9	202	304	9.58	48.52	5.064718	100
10	202	304	13.53	58.1	4.294161	150
11	202	304	20.4	89.4	4.382353	250
12	202	304	28.83	98.73	3.424558	350
13	202	304	39.53	101.33	2.56337	500

The results show that, out expectation of successive shortest path algorithm to be faster than cycle-cancelling algorithm is proven by the experiments. Successive shortest path algorithm runs 3 to 5 times faster than cycle-canceling algorithm. Also, as the number of nodes and arcs increase, both algorithms get slower as expected. Maximum demand has also a negative effect on the execution time of both algorithms. Lastly, as the maximum demand increases, speed difference between algorithms is also decreases.

5. Future Work & Recommendations

In the project, a simple idea of building evacuation models is used as an example. Synthetic networks are generated and solved to compare algorithms. However, there are more realistic models exist. As an example, Chalmet et al[2] represents more realistic approach to building evacuation models. By using more complex models and real-life examples of buildings, the work may create a practical value.

[2] Chalmet, L. G., R. L. Francis, and P. B. Saunders. "Network models for building evacuation." Management science 28.1 (1982): 86-105.

The algorithms used for solving problems are mostly the basic algorithms. We can increase the performance of algorithms by using more efficient ideas. For example, we can increase the performance successive shortest path algorithm by using Dijkstra's algorithm instead of Bellman-Ford. This may increase the speed significantly. Also, we can increase the performance by using more efficient data structures as well.

The implemented algorithms finds optimal solutions for feasible problems. By adding new rules, we may detect infeasibility and unboundedness, therefore, increase the capacilities of algorithms. Lastly, data entry and and output analysis processess can be designed in a more user-friendly manner. As a result, the implemented algorithms can be more efficiently used for solution of different problems.

Lastly, I was not able to compare the performance of my implementations with the commercial packages such as CPLEX due to some unresolved errors on linking CPLEX with Visual Studio. Therefore, the comparison can be done in the future works.

6. Conclusion

In this project, cycle-canceling and successive shortest path algorithms are implemented for solving building evacution model which can be modeled and solved as a minimum cost flow problem. Then, the performance of the algorithms are compared by running them on several problem instances. By doing this work, I gained the significant experience of implementation process. The theoretical background gained throughout the semester come into reality. By working on the implementation, I believe that I made my knowledge on the subject much more stronger then merely working on theoretical aspects.

7. References

1. Chalmet, L. G., R. L. Francis, and P. B. Saunders. "Network models for building evacuation." Management science 28.1 (1982): 86-105.
2. Ahuja, Ravindra K., Tomas L. Magnanti, and James B. Orlin. "Network flows. 1993."