Routing Optimization of Small Satellite Networks based on Multi-commodity Flow

Xiaolin Xu[1], Yu Zhang[2,*] and Jihua Lu[3]

[1] Beijing Institute of Technology, No.5 YardZhong Guan Cun South Street, Haidian District, Beijing
[2] Beijing Institute of Technology; The Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory, the 54th Research Institute of China Electronics Technology Group Corporation, No.5 YardZhong Guan Cun South Street, Haidian District, Beijing
[3] Beijing Institute of Technology, No.5 YardZhong Guan Cun South Street, Haidian District, Beijing

Abstract

As the scale of small satellite network is not large and the transmission cost is high, it is necessary to optimize the routing problem. We apply the traditional time-expanded graph to model the data acquisition of small satellite network so that we can formulate the data acquisition into a multi-commodity concurrent flow optimization problem (MCFP) aiming at maximizing the throughput. We use an approximation method to accelerate the solution for MCFP and make global optimization of routing between satellite network nodes. After the quantitative comparison between our MCFP algorithm and general augmented path maximum flow algorithm and exploring the detail of the algorithm, we verify the approximation algorithm's reasonable selection of routing optimization in small satellite network node communication.

1. Introduction

Recently, more and more small satellites are launched to carry out various space missions. Small satellites receive mission data from observable objects and send these data to the data processing center via ground stations or relay satellites [1]. As the scale of small satellite network is not large and the transmission cost is high, we need to optimize the routing and allocate the resources properly.

The time-expanded graph (TEG) is a useful tool to model the topology of network [2]. To deal with the challenges caused by the impacts of satellites' movements during delivery process, Runzi Liu and Min Sheng apply the traditional time-expanded graph to model the data acquisition [3]. The delivery strategies and the data acquisition are formulated into an optimization problem to maximize the throughput. For the tiny topology with few satellites, the problems of routing and resources allocation have been emerged as a topic on multi-commodity problem (MCP). However, Runzi Liu and Min Sheng concentrate on the small satellite model using TEG instead of giving a practical algorithm to solve it.

Though finding an integer flow solution to the MCP is proved to be an NP-complete problem [4]. A polynomial time solution has been found through carrying Linear Programming by allowing fractional flows [5]. Moreover, researchers have concentrated on approximation schemes to speed up the solution. Following by Young in 1995 [7], Shahrokhi and Matula proposed a new algorithm for maximum concurrent flow problem (MCFP) in 1990 [6], whose algorithm was improved by Garg and Konemann in 1998 [8]. Garg and Konemann simplified the ideas of Young and built a framework for computing the MCFP. After that, Fleischer realized that an approximation of which commodity has the shortest path could be made in finding a shortest path between the source-sink pairs and extended the framework in 2000 [9]. She was able to describe an algorithm and its running time is independent of the number of the commodity in the MCFP problem. We will use a modified version of Fleischer's algorithm to consider the problem.

In this paper, we first extend the TEG to put up the model of a small satellite topology. Our model builds on the framework proposed by Runzi Liu and Min Sheng [3]. Then we apply a polynomial time multi-commodity optimization algorithm to maximize the network

*Corresponding author. Email: yuzhang@bit.edu.cn

throughput based on this graph model. Simulation results highlight the practicality of our algorithm and explore the detail of our algorithm on parameter and running time et.al.

2. System model and optimal algorithm

2.1. System setup

We consider a Graph G, which represents a small satellite network (SSN). There are nodes $SN = \{s_1, s_2, ..., s_n, ...\}$ and $TN = \{t_1, t_2, ..., t_n, ...\}$ representing the source and destination of data and a number of missions $OM = \{om_1, om_2, ..., om_n, ...\}$ to be completed over these nodes, where $om_i = [s_i, t_i, d_i]$. Mission om_i comprises source nodes which connects observable object ob_i, destination nodes which connects data processing center dc_i and its demand d_i.

First we set our demands for each mission om_i. Small satellites which revolve around the earth at a low altitude of 350-1400 km acquire mission data when they moving into the coverage of the observable objects and the ground stations get the mission data via relay satellites or the small satellites directly in SSN. Then, the mission data is transmitted from the ground stations to the data processing center (DPC). As the small satellite can send mission data to a relay satellite or ground station only when it moves close to them due to the orbiting movements, the connectivity relationships between relay satellites (or ground stations) and small satellites are time varying. On the other hand, relay satellites locate on the geosynchronous orbit. That is, the connectivity relationships between relay satellites and ground stations are fixed.

The SSN consists:

•Source of data $SN = \{s_1, s_2, ..., s_n, ...\}$.

•Observable objects $OB = \{ob_1, ob_2, ..., ob_n, ...\}$.

•Small satellites $SS = \{ss_1, ss_2, ..., ss_n, ...\}$.

•Relay satellites denoted by $RS = \{rs_1, rs_2, ..., rs_n, ...\}$.

•Ground stations $GS = \{gs_1, gs_2, ..., gs_n, ...\}$.

•A data processing center, denoted by dc.

•Destination of data $TN = \{t_1, t_2, ..., t_n, ...\}$.

During each slot, the network topology of SSN is constant. But during slot transitions it could change instantaneously. We use TEG to capture the impact of satellites' orbiting movements on data acquisition. We divide the plan horizon $[0, T]$ into K slots with duration of $\tau = T / k$.

The TEG, denoted by $G = (V, E)$, is shown in Fig. 1. Here, $G = (V, E)$, is a directed graph that corresponds to a network topology with K slots. The vertices of G represent the copy of source nodes, destination nodes,

small satellites, observable objects, ground stations, relay satellites and data process centers for each slot. That is $V = V_S \bigcup V_{ob} \bigcup V_{ss} \bigcup V_{gs} \bigcup V_{rs} \bigcup V_{dc} \bigcup V_T$.

Graph G have five kinds of arcs: artificial arcs for SN and TN, data collection arcs, data storage arcs, fixed link arcs and opportunistic link arcs. We use the artificial arcs to accumulate the total transmission data and set the data rate of such links infinity, $C(dc_j^k, t_i) = \infty$, $C(s_i, ob_j^k) = \infty$. The data collection arcs represent capability of small satellites gathering mission data from observable objects, $E_{dc} = (ob_i^k, ss_j^k)$ and their capability equals the rate of mission data that small satellite ss_j can collect from observable object ob_i in the k_{th} slot,

$$C(ob_i^k, ss_j^k) = r_{dc} \qquad (1)$$

where r_{dc} is the data collection rate of small satellites. The data storage arcs correspond to the capability of satellites, stations and data process centers to store data, which is defined as $E_s = \{(v_i^k, v_i^{k+1}) \mid v_i^k \in V_{ss} \bigcup V_{rs} \bigcup V_{gs} \bigcup V_{dc}, 1 \leq k \leq K - 1\}$. We set the capacity of data storage arc (ss_i^k, ss_i^{k+1}) infinity, $C(ss_i^k, ss_i^{k+1}) = \infty$. Arcs $E_{fl} = \{(rs_i^k, gs_i^k) \mid 1 \leq i \leq RS, 1 \leq k \leq K\} \bigcup \{(gs_i^k, dc^k) \mid 1 \leq i \leq GS, 1 \leq k \leq K\}$ are fixed and link arcs denoted by $E_{ol} = \{(ss_i^k, vs_j^k) \mid ss_i^k \in V_{SS}, vs_j^k \in V_{rs} \bigcup V_{gs}, 1 \leq k \leq K\}$ are opportunistic. Their capacity is the rate that mission data is able to be sent by the link,

$$C(vt_i^k, vr_j^k) = r(vt_i, vr_j) \qquad (2)$$

where $r(vt_i, vr_j)$ is the data rate of link (vt_i, vr_j). Because high speed wired links connect the data process centers and ground stations, we can assume the data transmission rate of them are enough mass, that is $CD(gs_i^k, dc^k) = \infty$.
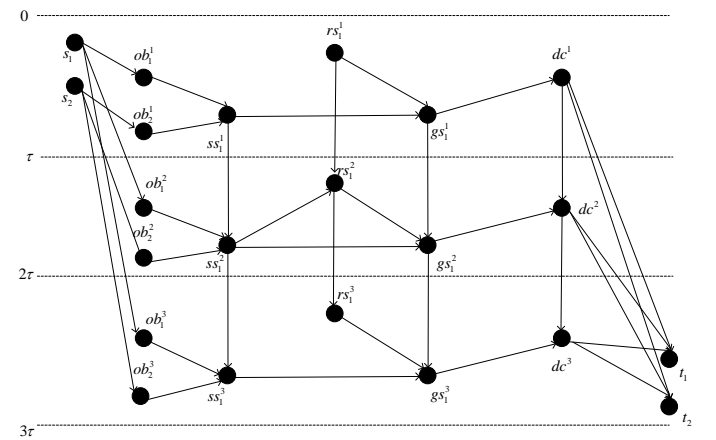

Figure 1. An example of extending TEG to network model

2.2. Multi-commodity Algorithm

As according to transformation before, the impact of network dynamics on delivery process can be modeled mathematically using the extended TEG as Fig.2. And the problem of routing optimization of small satellite network has been corresponding to a topic on a directed graph $G = (V, E)$ with k pairs of demands (s_j, t_j) $1 \le j \le k$ and the capacities of the edges are denoted by $u : E \to R$ which is equivalent to multi-commodity flow problem (MCP).
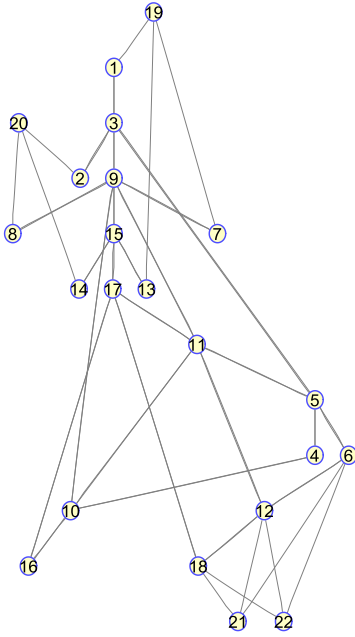

Figure 2. Graph corresponding to the extended TEG in Figure.1

For the MCP which concludes k source-sink demand pairs (s_j, t_j), as we use the rates to present the capacity of the edges, routing optimization of small satellite networks based on TEG come to a maximum concurrent flow problem (MCFP). The MCFP is a multi-commodity flow problem and all pairs of demands concurrently flow. For MCFP, the target is to assign flow to global route so that the ratio (termed the throughput) of the flow contributed between a pair equals to all pairs of demands. This assignment must not exceed the capacities of all the edges. Each commodity corresponds to a specified demand $d_j (1 \le j \le k)$ in MCFP. Finding a flow that maximizes ratio of demands is our purpose. Letting $x(P)$ denote the quantity of the flow on path P, MCFP can be formulated as:

$$
\begin{aligned}
\max \quad & \lambda \\
s.t. \quad & \sum_{P:e \in P} x(P) \le u(e) \quad \forall e \\
& \sum_{P \in P_j} x(P) \ge \lambda d_j \quad \forall j \\
& x(P) \ge 0 \quad \forall P
\end{aligned}
\tag{3}
$$

Polynomial solution for this problem can be found by using Linear Programming. To make the solution faster, we use a modified version of a different approximation algorithm from LISA K. FLEISCHER [9].

Letting $l(e) = \delta / u(e)$, $z_j = \min_{P \in P_j} l(P)$, $x \equiv 0$ at first. The entire procedure is in phases and there are k iterations in each phase. The goal is routing d_j units flow from s_j to t_j in iteration j in steps. We apply the Dijkstra's shortest path algorithm with the length function and computes the shortest path P from s_j to t_j in iteration j. The minimum of the remaining demand and the bottleneck capacity of this path will be transmitted on P. Then the length $l(e)$ renews and we set z_j the current minimum length of the path from s_j to t_j. The algorithm stops until the function value $D(l)$ is upper than one, that is $\sum_e u(e)l(e) > 1$. A summary of the algorithm in Figure.3, where we update the $l(e)$ by $1 + \varepsilon \frac{u}{u(e)}$ and set $\delta = (\frac{m}{1-\varepsilon})^{-1/\varepsilon}$.

Input: network G, capacities $u(e)$, vertex pairs (s_j, t_j) with demands d_i, $1 \le i \le k$, accuracy ε
Output: primal (infeasible) and dual solutions x and l

Initialize $l(e) = \delta / u(e)$ $\forall e$, $x \equiv 0$.
while $D(l) < 1$
  for $j = 1$ to k do
    $d_j' \longleftarrow d_j$
    while $D(l) < 1$ and $d_j' > 0$
      $P \longleftarrow$ shortest path in P_j using l
      $u \longleftarrow \min\{d_j', \min_{e \in P} u(e)\}$
      $d_j' \longleftarrow d_j' - u$
      $x(P) \longleftarrow x(P) + u$
      $\forall e \in P$, $l(e) \longleftarrow l(e)(1 + \frac{\varepsilon u}{u(e)})$
    end while
  end while
Return (x, l).

Figure 3. Algorithm for MCFP

The algorithm required no more than $2k \log m (\log k + \frac{1}{\varepsilon^2})$ iterations and the total time required by the ε-approximate solution is in $O^*(\varepsilon^{-2} m(k + m))$ time. If $D(l) >= 1$, then it will be sure that we can obtain at least $(1 - 3\varepsilon)$ times of the optimal solution by scaling the final flow by $\log_{1+\varepsilon} 1/\delta$.

3. Simulation

As the scale of small satellite network is not large, we define three scenarios of small satellites network to explore the influence of topologies and after extending TEG, there are respectively twenty-two, twenty-five and twenty-eight nodes in the graph. We assume that the capacities of all edges in the graph are twenty, except the infinity edges defined before and we set several source-sink demand pairs corresponding to specific demands. To investigate the effect of different algorithms, we choose a general method called augmented path maximum flow algorithm that is using single commodity max-flow algorithm for each source-sink demand pairs in sequence and augmenting the residual network graph every time after the single commodity max-flow is implemented. Then we will compare our MCFP algorithm with the augmented path maximum flow algorithm.
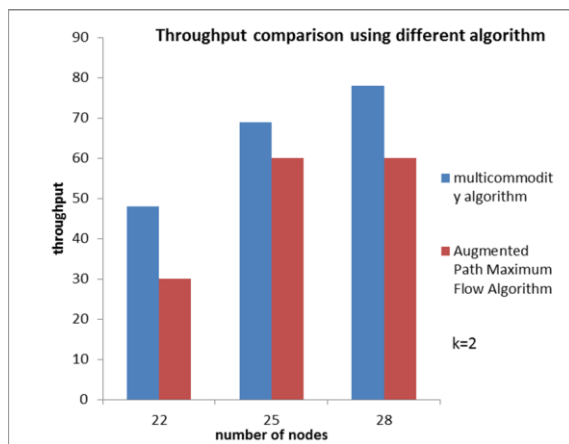

Figure 4. Results of MCFP algorithm and augmented path maximum flow algorithm (k=2)
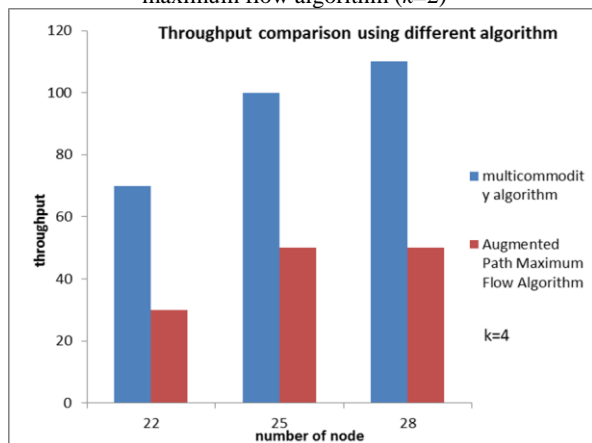

Figure 5. Results of MCFP algorithm and augmented path maximum flow algorithm (k=4)

Results of the two algorithms are shown in the Fig.4 and Fig.5. The augmented path maximum flow algorithm is able to obtain the maximum throughput between single source-sink nodes, it cannot collaboratively optimize the maximum flow path traffic for multiple source-sink node pairs and the optimization of routing performance cannot

be achieved. With different node numbers and different source-sink node pair requests, the small satellite network throughput of our method based on TEG model is obviously larger than that obtained by using the augmented path maximum flow algorithm. Furthermore, comparing the results of two source-sink demand pairs and four source-sink demand pairs, the difference between the two algorithms is more obvious with four source-sink demands. We can see that the more complicated the topology is as well as the more demands it has, MCFP algorithm will have better effects than general solution.
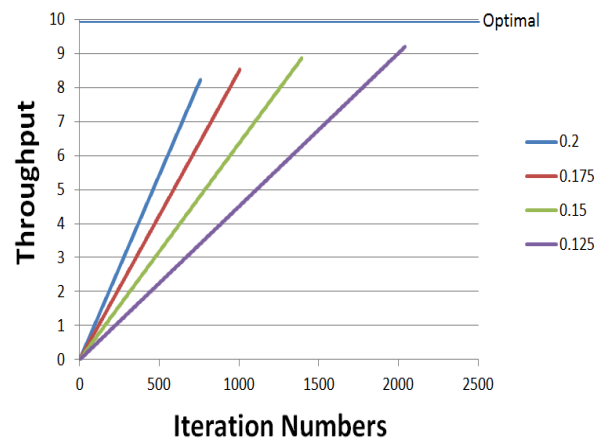

Figure 6. Throughput v.s. ε with 20 nodes (k=4)

Table 1. The change of ε

ε	Iterations	Time(s)	throughput
0.5	98	0.184	5.136
0.4	165	0.265	6.053
0.3	314	0.429	7.055
0.2	758	0.999	8.219
0.15	1395	1.672	8.861
0.1	3244	3.700	9.538

Both Fig.6 and Table.1 show the details for the algorithm to solve the TEG graph of a satellite network with 22 vertices and 36 edges.

To explore the dependency on ε, we use the algorithm to solve a flow network of small satellite where the optimal throughout (*OPT*) is 9.9. *OPT* is marked as the gray horizontal line at the top in Fig.6 and the closer to *OPT* ε is, the better results we get. The figure also shows the $(1-3\varepsilon)$ approximation guarantee of the solution. The guarantee says that the algorithm will produce a flow F such that $(1-3\varepsilon)OPT \leq F \leq OPT$, where F is the size of the flow. This guarantee means that the algorithm will produce a flow that its throughput is extremely close to *OPT*, in another word ,is above the guarantee line, and under the *OPT* line.

The values of ε, iterations, running time and throughput can be found in Table.1. To look in detail, as the number of ε decreases, the algorithm needs more iterations and time to get the solution and the final throughput will be more optimal. From another point of view, if the network changes fast and requires making decision quickly and in time, choosing a reasonable ε to achieve the balance of the running time and the accuracy is a wise choice.

4. Conclusion

We apply the traditional time-expanded graph to model the data acquisition of small satellite network and use an approximation method to accelerate the solution for MCFP and make global optimization of routing between satellite network nodes. The quantitative comparison between our MCFP algorithm and general augmented path maximum flow algorithm proves the algorithm achieving better throughput and being closer to the optimal solution. Exploring the detail of the algorithm, we conclude that we need to make a reasonable selection of parameter in our algorithm for satellite network nodes communication to achieve the balance of the running time and the accuracy.

References

[1] R. S. Jakhu and J. N. Pelton, Small Satellites and Their Regulation. New York, NY, USA: Springer, 2014.

[2] L. R. Ford and D. R. Fulkerson, Flows in Networks. Princeton, NJ, USA:Princeton Univ. Press, 1962.

[3] Runzi Liu, Student Member, IEEE, Min Sheng, Member, IEEE, King-Shan Lui, Senior Member, IEEE, Xijun Wang, Member, IEEE, Yu Wang, and Di Zhou, "An Analytical Framework for Resource-Limited Small Satellite Networks," IEEE COMMUNICATIONS LETTERS, vol. 20, no. 2, February. 2016.

[4] Shimon Even, Alon Itai, and Adi Shamir. On the complexity of time table and multi-commodity flow problems. In Foundations of Computer Science, 1975., 16th Annual Symposium on, pages 184–193. IEEE, 1975.

[5] Masao Iri. On an extension of the maximum-flow minimum-cut theorem to multicommodity flows. Journal of the Operations Research Society of Japan, 13(3):129–135, 1971.

[6] Farhad Shahrokhi and David W Matula. The maximum concurrent flow problem. Journal of the ACM (JACM), 37(2):318–334, 1990.

[7] Neal E Young. Randomized rounding without solving the linear program. In SODA, volume 95, pages 170–178, 1995.

[8] Naveen Garg and Jochen Koenemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. SIAM Journal on Computing, 37(2):630–652, 2007.

[9] Lisa K Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. SIAM Journal on Discrete Mathematics,13(4):505–520, 2000.

[10] Lester R Ford and Delbert R Fulkerson. Maximal flow through a network. Canadian journal of Mathematics, 8(3):399–404, 1956.