



A Novel Dynamic Multi-source Multi-sink Flow Algorithm over the Satellite Networks

Peng Wang, Hongyan Li^(✉), Tao Zhang, Shun Zhang, and Keyi Shi

State Key Laboratory of Integrated Service Networks, Xidian University,
Xi'an 710071, China
hyli@xidian.edu.cn

Abstract. The multi-source multi-sink maximum flow problem can be of great significance in guiding network optimization, service scheduling, and capacity analysis. With intermittent connectivity and time-dependence characteristics of satellite networks, the existing flow algorithms for multi-source multi-sink without temporal dimension involvement can no longer maintain high efficiency in satellite networks. To overcome the problem, we propose a novel dynamic multi-source multi-sink flow algorithm. Specially, the storage time-aggregated graph (STAG) is adopted to depict the time-varying properties of satellite networks. Then, a novel dynamic multi-source multi-sink flow (DMMF) algorithm is proposed to enhance the satellite networks' resource utilization. At last, the simulation is conducted, and the results with obvious network performance gain verifies our proposed DMMF algorithm.

Keywords: Storage time-aggregated graph ·
Dynamic multi-source multi-sink flow · Satellite networks

1 Introduction

The multi-source multi-sink maximum flow problem can effectively represent the maximum transmission ability of networks and assign data flow, and has great guiding value in network optimization, capacity analysis and service scheduling [2]. However, with the time-varying topology and no persistent paths for source-destination nodes in satellite networks [3], it will be more difficult to find efficient dynamic multi-source multi-sink flow in satellite networks with temporal-dimension involved. As shown in Fig. 1, without precise description of time-varying satellite network resources (i.e. transmission resources, storage resources), static-graph based maximum flow algorithm can no longer keep efficient [1, 4, 5]. To be more specific, with the network topology shown in Fig. 1 and $\{\{S, D\}, \{A, B\}\}$ being the two source-sink node pairs, the maximum flow calculated by the static-graph based algorithm [4] for $\{S, D\}$ is 3 and for $\{A, B\}$ is 0. In fact, with storage

Supported by organization x.

resources involved and considerate analysis of the relationship between the two flows, the maximum flow for $\{S, D\}, \{A, B\}$ can be 3 respectively.

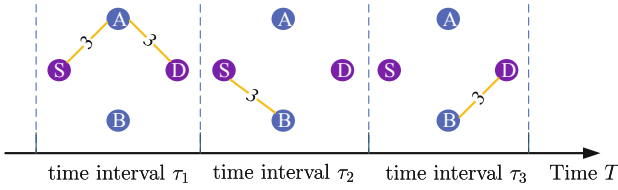


Fig. 1. A simple illustration of satellite network topology within 3 time intervals

Hence, we will propose a novel dynamic multi-source multi-sink algorithms for satellite networks to enhance the transmission capacity of satellite networks. First, the storage time aggregated graph based on our previous work [7] is adopted to depict the network topology of satellite networks with temporal-dimension involvement. Then the feasibilities of flows with different source-node pairs are carefully considered and the special constraints are derived. Then based on the dynamic two-commodity maximum combined flow in our previous work [9], we propose the heuristic novel DMMF algorithm, which can effectively increase the network transmission ability among different source-sink node pairs compared with the existing algorithms. For completeness and clearness, one virtual satellite network with time-vary topology is constructed, and the performance of our proposed DMMF algorithm is verified by comparing with other flow assignment algorithms.

2 Satellite Networks Model

With the trajectories of satellites predictable, the communication windows among satellites could be precisely obtained. Based on the predictable feature and reasonable assumptions, we adopted storage-time aggregated graph [7] to model the time-varying resources (i.e. satellites' storage resources, communication windows, data transmission capacity of links) of satellite networks.

Specifically, given time horizon $T = [t_0, t_n]$, and the specific satellites which the satellite networks consist of, the communication windows between satellites can be predicted from STK. With the communication time windows among satellites obtained and reasonable assumptions, the undirected $STAG = \{T, \mathcal{V}, \mathcal{E}, \mathcal{C}_{u,v}(T), \mathcal{N}_v(T), \mathcal{S}_{\mathcal{V}} | u \in \mathcal{V}, v \in \mathcal{V}, (u, v) \in \mathcal{E}\}$ is assorted to model the satellite networks:

- $T = [t_0, t_n]$, a time horizon of the satellite networks, which is not the life time of satellite networks. We split $[t_0, t_n]$ into n small time intervals $\{\tau_1, \dots, \tau_p, \dots, \tau_n\}$, where $\tau_p = (t_{p-1}, t_p]$ and during each small time interval τ , the topology of the satellite network is fixed.

- \mathcal{V} , the set of the nodes (satellites).
- \mathcal{E} , the set of undirected edges, where $(u, v) \in \mathcal{E}$ holds if and only if the contact time window between satellite u and v during T is nonzero.
- $\mathcal{C}_{u,v}(T) = \{c_{u,v}(\tau_1), \dots, c_{u,v}(\tau_p), \dots, c_{u,v}(\tau_n)\}$, is a capacity time series set for each link $(u, v) \in \mathcal{E}$, where $c_{u,v}(\tau_p) = \int_{t_{p-1}}^{t_p} w_{u,v}(t)dt$, and $w_{u,v}(t)$ depict the link capacity of edge (u, v) at time instant t . Note that even if links of satellite works are not directional, flows themselves are directional.
- $\mathcal{N}_v(T) = \{N_v(\tau_1, \tau_2), \dots, N_v(\tau_p, \tau_{p+1}), \dots, N_v(\tau_{n-1}, n)\}$, a node storage time series set, where $N_v(\tau_p, \tau_{p+1})$ records the storage resources utilization of each node $v \in \mathcal{V}$ from τ_p to τ_{p+1} . And the storage resources correspond to maximum amount of data one node can store from one small time interval to the adjacent one.
- $\mathcal{S}_\mathcal{V}$, a storage resource set, where the storage resources of each satellite $v \in \mathcal{V}$ at time instant t_0 is denoted as s_v .

To be more explicit, one simple satellite network (S, D, B, A) within 3 time intervals is depicted by undirected STAG shown in Fig. 2 as further illustration.

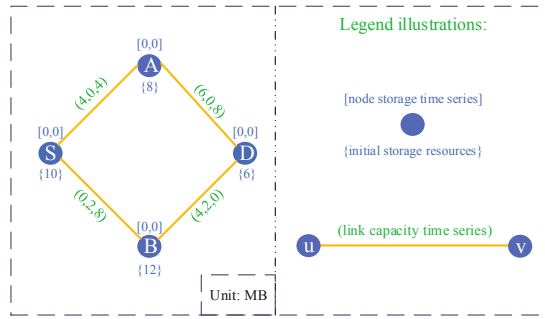


Fig. 2. Satellite network modeled by undirected STAG within 3 time intervals

3 Feasible Multi-source Multi-sink Flow Definitions in STAG

Unlike the static flow definitions with the links' directed capacity determined and just considered in networks with fixed topology, there are some remarkable differences here for the feasible and dynamic multi-source multi-sink flow in the undirected $STAG = \{T, \mathcal{V}, \mathcal{E}, \mathcal{C}_{u,v}(T), \mathcal{N}_v(T), \mathcal{S}_\mathcal{V} | u \in \mathcal{V}, v \in \mathcal{V}, (u, v) \in \mathcal{E}\}$ without loops and parallel edges. With $\{S_k, D_k\}$ being the k_{th} source-sink node pair, then the flow of the k_{th} source-sink node pair from u to v within T can be noted as:

$$f_{u,v}^k(T) = \sum_{q=1}^n f_{u,v}^k(\tau_q), \tag{1}$$

where $f_{u,v}^k(\tau_q)$ means the flow of k_{th} source-sink nodes from u to v within τ_q .

Then with the h source-sink nodes given as $\{(\mathbb{S}_1, \mathbb{D}_1), \dots, (\mathbb{S}_k, \mathbb{D}_k), \dots, (\mathbb{S}_h, \mathbb{D}_h)\}$, the total flow of the h source-sink nodes is:

$$sum = F_{\mathbb{S}_1, \mathbb{D}_1}^1(T) + \dots + F_{\mathbb{S}_k, \mathbb{D}_k}^k(T) + \dots + F_{\mathbb{S}_h, \mathbb{D}_h}^h(T), \tag{2}$$

where the total flow of the k_{th} source-sink nodes is:

$$F_{\mathbb{S}_k, \mathbb{D}_k}^k(T) = \sum_{v \in \mathbb{V}} f_{\mathbb{S}_k, v}^k(T) - \sum_{v \in \mathbb{V}} f_{v, \mathbb{S}_k}^k(T). \tag{3}$$

With the definitions given above, the specific constraints for h source-sink node pairs in the undirected STAG should be satisfied as follows:

Capacity Constraints: Note that the speciality of the undirected graph is that flow assignment of each source-sink node pairs between two nodes can only be single directed. To be more explicit:

$$|f_{u,v}^k(\tau_p)| \times |f_{v,u}^k(\tau_p)| = 0, \forall p \in [1, n], \forall k \in [1, h], \tag{4}$$

where the flow assignment $f_{u,v}^k(\tau_p)$ and $f_{v,u}^k(\tau_p)$ should be non-negative. And with the direction rule of each source-sink flow assignment, in undirected STAG:

$$0 \leq \sum_{k=1}^h f_{u,v}^k(\tau_p) + \sum_{k=1}^h f_{v,u}^k(\tau_p) \leq c_{u,v}^{\tau_p}, \forall p \in [1, n]. \tag{5}$$

Node Storage Transfer Constraints: For k_{th} source-sink determined flow ($\forall k \in [1, h]$), within τ_1 :

$$\sum_{v \in \mathbb{V}} f_{v,u}^k(\tau_1) - \sum_{v \in \mathbb{V}} f_{u,v}^k(\tau_1) = N_u^k(\tau_1, \tau_2). \tag{6}$$

Naturally, during the next time intervals $\forall p \in [2, n - 1]$:

$$\sum_{v \in \mathbb{V}} f_{v,u}^k(\tau_p) + N_u^k(\tau_{p-1}, \tau_p) - \sum_{v \in \mathbb{V}} f_{u,v}^k(\tau_p) = N_u^k(\tau_p, \tau_{p+1}). \tag{7}$$

With simple mathematical operations, the storage transfer constraints within T can be easily obtained as:

$$\sum_{p=1}^q \sum_{v \in \mathbb{V}} f_{v,u}^k(\tau_p) - \sum_{p=1}^q \sum_{v \in \mathbb{V}} f_{u,v}^k(\tau_p) = \begin{cases} N_u^k(\tau_p, \tau_{p+1}), q \in [1, n - 1]. \\ 0, q = n. \end{cases} \tag{8}$$

Storage Resource Constraints: And with the constraints of each source-sink flow satisfied, there exists the following storage resource constraints:

$$\begin{cases} 0 \leq N_v^k(\tau_p, \tau_{p+1}) \leq s_v, \forall k \in [1, h], \forall p \in [1, n - 1]. \\ 0 \leq \sum_{k=1}^h N_v^k(\tau_p, \tau_{p+1}) \leq s_v, \forall p \in [1, n - 1]. \end{cases} \tag{9}$$

Flow Conservation: Then for each source-sink node pair determined flow, there exists the following flow conservation:

$$\sum_{v \in \mathcal{V}} f_{u,v}^k(T) - \sum_{v \in \mathcal{V}} f_{v,u}^k(T) = \begin{cases} F_{\mathbb{S}_k, \mathbb{D}_k}^k(T), u = \mathbb{S}_k, \\ 0, u \neq \mathbb{S}_k, \mathbb{D}_k, \\ -F_{\mathbb{S}_k, \mathbb{D}_k}^k(T), u = \mathbb{D}_k. \end{cases} \quad (10)$$

With the above constraints for feasible DMMF in undirected STAG, it's more complicated to obtain a rather-efficient feasible DMMF in undirected STAG compared with the static graph. Mechanical re-conduct of time-varying maximum flow solution can result in severe network resources waste.

4 Heuristic Multi-source Multi-sink Flow Algorithm

The multi-commodity flow algorithm is a NP-hard problem and has been proved in [6], and it will be more difficult to solve in the disruption tolerant satellite networks. To enhance the transmission ability in satellite networks, we have solved the dynamic maximum single-commodity [7, 8], two-commodity flow problem [9] in our previous works. To overcome the problem of efficient dynamic multi-source multi-sink flow assignment problem, a heuristic DMMF algorithm is proposed

Algorithm 1. The Heuristic Multi-source Multi-sink Flow Algorithm

- 1: **Input:** STAG = $\{T, \mathcal{V}, \mathcal{E}, \mathcal{C}_{u,v}(T), \mathcal{N}_v(T), \mathcal{S}_{\mathcal{V}} | u \in \mathcal{V}, v \in \mathcal{V}, (u, v) \in \mathcal{E}\}$, h source-sink node pairs $\{(\mathbb{S}_1, \mathbb{D}_1), \dots, (\mathbb{S}_k, \mathbb{D}_k), \dots, (\mathbb{S}_h, \mathbb{D}_h) | h > 2\}$.
 - 2: **Output:** The feasible efficiently dynamic flow assignment of each source-node pair.
 - 3: **Decompose** the original problem of solving the dynamic and efficient feasible flow for h source-sink nodes over the satellite networks into two-commodity maximum dynamic flow problem by inserting two virtual source-sink nodes pairs $\{(\mathbb{S}'_1, \mathbb{D}'_1), (\mathbb{S}'_2, \mathbb{D}'_2)\}$ to \mathcal{V} .
 - 4: **Define** j as an integer counter and the initial value of j is 1.
 - 5: **while** $j \leq \lfloor h/2 \rfloor$ **do**
 - 6: plug virtual link $(\mathbb{S}'_1, \mathbb{S}_j)$ and $(\mathbb{D}_j, \mathbb{D}'_1)$ into \mathcal{E} .
 - 7: $C_{\mathbb{S}'_1, \mathbb{S}_j}(T) = \sum_{v: (\mathbb{S}_j, v) \in \mathcal{E}} C_{\mathbb{S}_j, v}(T)$, $C_{\mathbb{D}_j, \mathbb{D}'_1}(T) = \sum_{v: (\mathbb{D}_j, v) \in \mathcal{E}} C_{\mathbb{D}_j, v}(T)$.
 - 8: increase j by 1.
 - 9: **end while**
 - 10: **while** $j \leq h$ **do**
 - 11: plug virtual link $(\mathbb{S}'_2, \mathbb{S}_j)$ and $(\mathbb{D}_j, \mathbb{D}'_2)$ into \mathcal{E} .
 - 12: $C_{\mathbb{S}'_2, \mathbb{S}_j}(T) = \sum_{v: (\mathbb{S}_j, v) \in \mathcal{E}} C_{\mathbb{S}_j, v}(T)$, $C_{\mathbb{D}_j, \mathbb{D}'_2}(T) = \sum_{v: (\mathbb{D}_j, v) \in \mathcal{E}} C_{\mathbb{D}_j, v}(T)$.
 - 13: increase j by 1.
 - 14: **end while**
 - 15: **Compute** the logically constructed two-commodity maximum flow assignment with **Algorithm 2**.
 - 16: **while** the flow assignment of each source-sink nodes is not feasible **do**
 - 17: Repeat this algorithm with input and output updated.
 - 18: **end while**
-

based on the efficiency of the maximum dynamic two-commodity flow, which can be shown in Algorithm 1.

With respect to the dynamic two-commodity maximum flow DCF algorithm, adopted in the same undirected STAG and with extra storage resource constraints involvement, the DCF algorithm can be perfectly transformed here to solve the efficient DMMF assignment problem as following. For completeness and clarity, the maximum dynamic two-commodity algorithm is presented as Algorithm 2. And the detailed construct, perform and adjust process and the corresponding proving process in Algorithm 2 can be found in our previous work [9].

The overall solution for efficient DMMF flow algorithm is to recursively decompose multi-source multi-sink flow problem into logically dynamic two-commodity maximum flow problem.

Algorithm 2. Maximum Dynamic Two-commodity Flow Algorithm

- 1: **Input:** $STAG = \{T, \mathcal{V}, \mathcal{E}, \mathcal{C}_{u,v}(T), \mathcal{N}_v(T), \mathcal{S}_v | u \in \mathcal{V}, v \in \mathcal{V}, (u, v) \in \mathcal{E}\}$, 2 source-sink node pairs $\{(\mathbb{S}_1, \mathbb{D}_1), (\mathbb{S}_2, \mathbb{D}_2)\}$.
 - 2: **Define** $f_{u,v}^+(\tau_q)$ the assignment of addition-flow from node u to node v within τ_q . Naturally, the assignment of minus-flow from node u to node v during τ_q is $f_{u,v}^-(\tau_q)$. Besides, For each node v , denote the storage transfer series in addition-flow graph as $N_v^+(T)$, and $N_v^-(T)$ records the storage transfer series in minus-flow graph.
 - 3: **Construct** the addition-flow graph $STAG^+$ and minus-flow graph $STAG^-$, based on which the coupling relationship of two commodity flow can be reduced into two independent ones by setting the single source-sink node pair of the two graphs as $(\{\mathbb{S}_1, \mathbb{S}_2\}, \{\mathbb{D}_1, \mathbb{D}_2\})$ and $(\{\mathbb{S}_1, \mathbb{D}_2\}, \{\mathbb{S}_2, \mathbb{D}_1\})$, respectively.
 - 4: **Perform** the STAG-based maximum flow algorithm in addition-flow graph and minus-flow graph respectively, and note that augmentation path search in $STAG^-$ is different from that in $STAG^+$, which can be explained as follows, to obtain the maximum flow in $STAG^-$, in addition to the normal augmenting flow process from source to sink node, augmenting flow from sink to source node is also necessary.
 - 5: **Adjust** the flow through the nodes in addition-flow graph and minus-flow graph respectively to satisfy the constraint (9).
 - 6: **Output** the nodes' and links' storage and capacity resources allocation for each source-node flow. Explicitly, for $\forall v \in \mathcal{V}$, the storage assignment for $F_{\mathbb{S}_1, \mathbb{D}_1}^1(T)$ is $(N_v^+(T) + N_v^-(T))/2$, and $F_{\mathbb{S}_2, \mathbb{D}_2}^2(T)$ is $(N_v^-(T) - N_v^+(T))/2$. Furthermore, for $\forall (u, v) \in \mathcal{E}$, the assignment for $f_{u,v}^1(T)$ is $(f_{u,v}^+(T) + f_{u,v}^-(T))/2$, and $f_{u,v}^2(T)$ is $(f_{u,v}^+(T) - f_{u,v}^-(T))/2$.
-

5 Simulation

To further verify the performance of our proposed dynamic multi-source multi-sink flow assignment algorithm, we consider one virtual satellite network with 12 satellites and 3 source-sink node pairs during an hour, and the length of each small time interval τ is set as 20 min, which can be seen from Fig. 3(a). Note

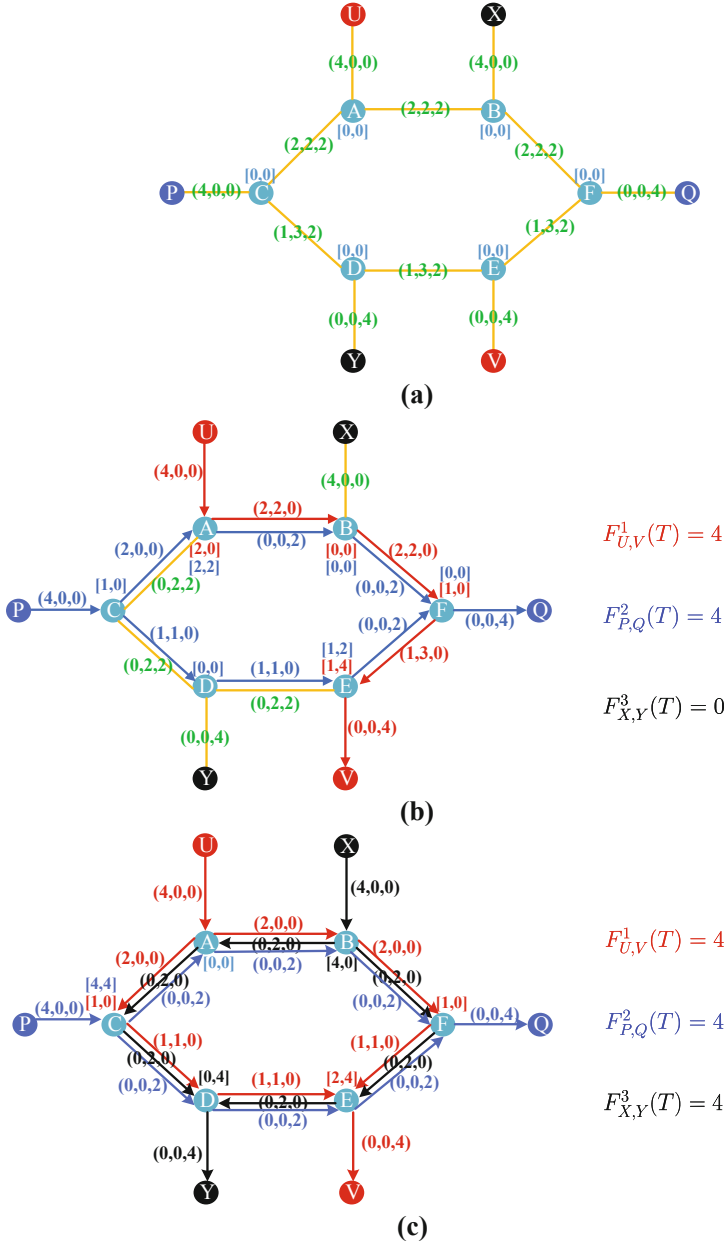


Fig. 3. (a) Topology of the virtual constructed network. (b) Flow assignment of the priority-based single commodity maximum flow algorithm. (c) Flow assignment of our DMMF.

that the network is virtual and constructed based on the time-varying characteristics. As shown in Fig. 3(b), the trivial solution for multi-source multi-sink flow problem is to repeat applying the STAG-based temporal single-commodity algorithm, and the flow result is 8 and there is no flow for the third commodity (X, Y) . And with our two-commodity based DMMF algorithm, the flow assignment can increase to 12, and the flow assignment for (X, Y) is also 4, which can be explained as our algorithm involved the coupling relationship between different source-sink node pairs. To be more explicit, the link and node resources are assigned jointly among multi source-sink nodes.

6 Conclusion

In this paper, we propose a STAG-based DMMF for satellite networks to efficiently assign the network flow and increase resource utilization. We construct one effective algorithm of multi-source multi-sink flow assignment, which is based on the efficiency of the undirected STAG and the dynamic maximum two-commodity flow algorithms. With the feasibility of flow assignment carefully considered, we decompose the multi-source multi-sink flow assignment problem into maximum dynamic two-commodity flow problem over the time-varying satellite networks.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (61871456, 91638202, 61401326, 61571351), the National Key Research and Development Program of China (2016YFB0501004), the National S&T Major Project (2015ZX03002006), the 111 Project (B08038), Natural Science Basic Research Plan in Shaanxi Province of China (2016JQ6054).

References

1. Borradaile, G., Klein, P.N., Mozes, S., Nussbaum, Y., Wulff-Nilsen, C.: Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, pp. 170–179, October 2011. <https://doi.org/10.1109/FOCS.2011.73>
2. Borradaile, G.: Planar maximum flow: multiple-source multiple-sink maximum flow in directed planar graphs (2015)
3. Caini, C., Cruickshank, H., Farrell, S., Marchese, M.: Delay- and disruption-tolerant networking (DTN): an alternative solution for future satellite networking applications. *Proc. IEEE* **99**(11), 1980–1997 (2011)
4. Ford, L.R., Fulkerson, D.R.: *Flows in Networks*. Princeton University Press, Princeton (1962)
5. Goldberg, A.V., Tarjan, R.E.: Efficient maximum flow algorithms. *Commun. ACM* **57**(8), 82–89 (2014)
6. Hall, A., Hippler, S., Skutella, M.: Multicommodity flows over time: efficient algorithms and complexity. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) *ICALP 2003*. LNCS, vol. 2719, pp. 397–409. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45061-0_33

7. Li, H., Zhang, T., Zhang, Y., Wang, K., Li, J.: A maximum flow algorithm based on storage time aggregated graph for delay-tolerant networks. *Ad Hoc Netw.* **59**, 63–70 (2017)
8. Wang, P., Zhang, X., Zhang, S., Li, H., Zhang, T.: Time-expanded graph based resource allocation over the satellite networks. *IEEE Wirel. Commun. Lett.* 1 (2018). <https://doi.org/10.1109/LWC.2018.2872996>
9. Zhang, T., Li, H., Li, J., Zhang, S., Shen, H.: A dynamic combined flow algorithm for the two-commodity max-flow problem over the delay tolerant networks. *IEEE Trans. Wirel. Commun.* 1 (2018). <https://doi.org/10.1109/TWC.2018.2872551>