Received: 13 February 2017 | Revised: 12 March 2017 | Accepted: 11 April 2017
DOI: 10.1002/sat.1212

WILEY

RESEARCH ARTICLE

Time division inter-satellite link topology generation problem: Modeling and solution

Xiaogeng Chu | Yuning Chen

College of Information Systems and Management, National University of Defense Technology, Changsha 410073, Hunan, China

Correspondence
Yuning Chen, College of Information Systems and Management, National University of Defense Technology, Changsha 410073, Hunan, China.
Email: cynnudt@hotmail.com

Funding information
National Natural Science Foundation of China, Grant/Award Number: 71201171

Summary

In this paper, we study the time-division inter-satellite link topology generation (TDILTG) problem for the well-known Chinese BeiDou Global Navigation Satellite System. The TDILTG problem consists in generating a time-division topology of the inter-satellite link network for the navigation satellite system to spread systematic data to all satellites via a few source satellites with the purpose of minimizing the time required to spread the data. We propose a mathematical model to formulate the TDILTG problem and study its 2 lower bounds through a thorough analysis of the problem characteristics. We also present a deterministic constructive (DC) algorithm to solve this problem approximately but very quickly, with a time complexity of $O(n^3)$, where n is the number of satellites. Extensive experimental studies on a wide range of randomly generated instances show that the proposed DC algorithm is able to obtain the optimal solutions for most tested instances in less than 1 second. Meanwhile, we also validate that the DC algorithm performs well when the problem scale is large. Furthermore, we provide insights of the effects of different instance parameters on the final results.

KEYWORDS

combinatorial optimization, global navigation satellite system, heuristics algorithm, topology generation

1 | INTRODUCTION

One of the most important issues for the inter-satellite link network management of the well-known Chinese Global Navigation Satellite System[1] is how to generate a topology of the inter-satellite link network in a time-division multiplexing transmission mode. This leads to an interesting optimization problem that we named the time-division inter-satellite link topology generation (TDILTG) problem. The purpose of this problem is to generate a topology that minimizes the period of time required for spreading data[1] while satisfying a set of transmission constraints.

To make a low-power and large-scale inter-satellite link network,[2] the global Navigation Satellite System uses the time-division transmission mode to establish inter-satellites links, where the time horizon is divided into several slots. In each slot, each satellite connects to another satellite, and each satellite switches to another satellite in the next slot. It is thus appropriate to use an evolving graph to represent the topology of the inter-satellite link network, where the edges of the graph evolve over time. In the meantime, ground stations need to transmit systematic data to all satellites via limited source satellites, and each satellite is expected to receive the data as soon as possible.

Furthermore, each satellite in a pair of satellites having a connection must remain visible to the other within the effective time domain of the topology. Because the visibility relation of 2 satellites may change over time because of the relative motion of the satellites, the navigation system needs to generate the topology of the network every once in a while. It is thus necessary for the system to solve the problem automatically and effectively to meet the demands of engineering.

In research, the time-division topology of an inter-satellite link network is a kind of graph evolving over time[3]; this approach is a hot topic in computer science and widely used in dynamic network modeling.[4-7] The evolving graph has been used to monitor the connectivity of opportunistic sensor networks.[8] Additionally, the frequent change in the network topology has been found to be a potential challenge.[9,10] Furthermore, it has been proposed that the evolving graph can help capture the dynamic behavior of dynamic networks.[11-13] Evolving graphs have been used to capture the dynamic characteristics of such networks and thus to compute multicast trees with minimum overall transmission time for a class of wireless mobile dynamic networks.[13] The expected time needed to visit all vertices of such a dynamic graph has been studied[14] under the assumption that the graph is being modified by an oblivious adversary. Some approaches,[15] such as cSTAG, address the problem of finding clusters that optimize both temporal and spatial distance measures simultaneously. The speed of information spread[16] in the stationary phase has been widely studied by analyzing the completion time of the flooding mechanism, and bounds on the completion time of the flooding mechanism have been provided. Previous works such as Grindrod and Higham[17] and Clementi et al[18] have dealt only with the information spread given the topology of an evolving graph.

One notices that all the cited recent research has focused on the time required for information spread given the topology of the evolving graph. To the best of our knowledge, there is little research on the topology generation problem of an evolving graph, and the present study is an attempt to investigate it. This paper focuses on the need to generate the topology of the evolving graph representing the inter-satellite links of the navigation system aiming to broadcast a piece of information from several source nodes to all nodes as soon as possible. The paper makes 4 main contributions:

1. a proposal of a mathematical model of the TDILTG problem;
2. an analysis of the lower bound of the problem;
3. the design of a heuristic algorithm that can be used to solve the problem effectively; and
4. the generation of test instances and the reporting of experiments that validate the effectiveness of the proposed heuristic algorithm.

In the following sections, after a statement of the problem, various aspects of the problem are investigated. A mathematical model is first proposed, and the lower bound of the problem is then studied. Finally, a heuristic algorithm that solves the model is produced and verified with sufficient examples.

2 | MATHEMATICAL MODEL OF THE TDILTG PROBLEM

This section first introduces the problem description and the mathematical model of the TDILTG problem.

2.1 | Problem description

The Chinese Global Navigation Satellite System contains about 27 satellites, and the ground stations need to periodically spread the systematic data to all the satellites via the ground-satellite links and the inter-satellite links. The systematic data (mainly composed of time calibration information) need to be transmitted to all the satellites as soon as possible. Some domestic satellites that are visible to the ground stations are selected as the source satellites, which obtain the systematic data from the stations and forward them to all the other satellites via the inter-satellite links.

Inter-satellite links are established using a time-division transmission mode, which allows reducing the energy of satellites and increasing the number of satellites that can be connected. In the time-division transmission mode, there are 3 time-related concepts: time domain, time horizon, and time slot, whose relations are as follows. A “time domain” of the topology contains a number of time horizons, which use the same time division inter-satellite link topology. A “time horizon” is divided into a number of “time slots,” where in each of them, a satellite is allowed to establish a full duplex communication link with only one other visible satellite for data transmission.

The amount of the systematic data is small enough to be fully transmitted in a single slot through the established inter-satellite link, and the time horizon is long enough to ensure that the data can reach all satellites before the end of the time horizon. The topology of the network is kept unchanged within a time domain, but it needs to be regenerated in a new time domain because the visibility relation of 2 satellites may change over time because of the relative motion of the satellites.

In the light of graph theory, the visibility relationship of the satellites in one time domain of the topology could be defined as an undirected graph $G = (\mathrm{sat}, \mathrm{adj})$ where sat is a set of vertices that represent all the satellites and the adj is the adjacency matrix of the visibility relationship among all satellites ($\mathrm{adj}_{ij} = 1$ if satellite i is visible to satellite j). On the other hand, each satellite is required to receive the data before the end of the time horizon. It is ensured that the graph $G = (\mathrm{sat}, \mathrm{adj})$ is always connected.

To highlight the main point of the study, we focus on the data transmission path between the factories and each satellite when generating a topology of the network. Before proposing the mathematical model, we list a collection of basic assumptions that should hold for a transmission scheme.

1. The interval of the topology can be divided into several slots in which the connection relation of the network is stationary.
2. The quantity of systematic data is small, and the data can be transmitted completely through the connection in a single slot.
3. In the first slot, the stations send the systematic data to all the source satellites via satellite-ground links.
4. Any uninformed satellite becomes informed in slot t if and only if it establishes a connection with another informed satellite in slot t.
5. The visibility relation and transmission connection of 2 satellites is bidirectional.

2.2 | Details of the model

In the process of the data distribution of the satellites, ground stations send data to source satellites using the ground antenna in the first slot, and other satellites are used as the destination nodes. The purpose of the design of the topology is to ensure all satellites receive the systematic data as early as possible while satisfying several constraints. The inputs of the problem are the source satellites and the visibility relation matrix of satellites. The notations of the model are as follows.

1. C_i is the receiving slot of satellite i in which satellite i receives the data for the first time;
2. n is the total number of satellites;
3. T_{max} is the maximum number of time slots in the time horizon;
4. adj is the adjacency matrix of the visibility relationship among all satellites and is constant during a period, where $adj_{ij} = 1$ if satellite i is visible to satellite j and vice versa, and $adj_{ij} = 0$ otherwise;
5. S is the set of source satellites that can receive data via the satellite-ground links in the first slot and then send the data to all other satellites through the inter-satellites links;
6. t is the t-th slot of the topology of the inter-satellite link network; and
7. x_{ijt} is the decision variable of the topology, taking a value of 1 if satellite i establishes a connection with satellite j in the t-th slot $x_{ijt} = 1$ and $x_{ijt} = 0$ otherwise.

Given the source satellites and the visibility relation matrix (adjacency matrix) of satellites, the TDILTG problem can be formalized as Equations 1 to 9.

The objective of the TDILTG problem is

$$\text{Minimize} \max_{i \in \{1,\ldots,n\}} C_i \tag{1}$$

Equation 1 states that the objective is to minimize the data receiving time of the last satellite.

The constraints are

$$x_{ijt} \leq adj_{ij}, \forall i, j, t \tag{2}$$

$$x_{ijt} = x_{jit}, \forall i, j, t \tag{3}$$

$$\sum_{j \in \{1,\ldots,n\} \setminus \{i\}} x_{ijt} \leq 1, \forall i, t \tag{4}$$

$$C_i = 1, \forall i \in S \tag{5}$$

$$C_i = \min_t \left(t \times \sum_{j \in \{1,\ldots,n\} \{i\}} x_{ijt} \right), \quad \forall i \notin S \tag{6}$$

$$1 \leq C_i \leq T_{max}, \forall i \tag{7}$$

$$\min(C_i, C_j) \times x_{ijt} < M, \forall i, j, t \tag{8}$$

$$x_{ijt} \in \{0, \ 1\}, i \in \{1, \ldots, n\}, j \in \{1, \ldots, n\}, t \in \{1, \ldots, T_{max}\}. \tag{9}$$

Equations 2 to 9 give the constraints of the problem. Equation 2 states that 2 satellites can create a connection link only if they are mutually visible. Equation 3 states that the data are transmitted bidirectionally through the link. Equation 4 states that the number of links maintained by each satellite is at most one in any slot. Equation 5 states that the source nodes receive the data in the first slot. Equation 6 states that the receiving slot of any satellite, except for the source nodes, is the first slot in which the satellite establishes a connection with an informed satellite. Equation 7 states that every satellite has to receive the data before the end of the time horizon. Equation 8 states that at least 1 of the 2 satellites in the establishment of an inter-satellite link has the systematic data. Equation 9 determines the ranges of the variables. Subscripts i and j are indices of the satellites. The slot index is larger than one because the source satellites receive data in the first slot.

Theoretically, the number of satellites can be quite large, which implies a huge search space. Moreover, the presence of a set of hard constraints (like those quadratic ones) makes the problem difficult to solve.

3 | LOWER BOUND

This section obtains 2 lower bounds of the optimization problem through the analysis of the TDILTG problem. The first lower bound is acquired by computing the minimum-hop transmission path of each satellite. The second is acquired under the hypothesis of the complete graph in which any pair of nodes has an edge. To facilitate the introduction, the transmission hop count of each satellite is defined as the hop count computed by the minimum-hop routing algorithm under the adjacency matrix of the visibility relationship among all satellites.

3.1 | Lower bound 1

Let T be the number of slots needed for the spread of the data, and MHC be the hop count of the satellite that has the maximal transmission hop count from the source satellites, referred to as the shortest distance from the source for convenience.

Proposition 1.

$$T \geq MHC \tag{10}$$

Proof: *Because the data needs a slot to be transferred from one satellite to another, the minimum number of slots needed to send data to each satellite cannot be less than the transmission hop count of any satellite as defined above. It is thus concluded that the objective function is larger than the maximal transmission hop count of the satellites under the visibility relation matrix of all the satellites.*

In Figure 1, {sat1,sat2} denotes the set of source satellites, and there is an edge between 2 satellites if they are visible to each other. In Figure 1 A, the minimum-hop transmission path of sat6 is obviously {fac-sat2-sat4-sat5-sat6}, and the transmission hop count of sat6 is thus 4. In Figure 1B, the bold edges with a number indicate the connections of the optimal topology, and the number on the edge gives the order of the slot in which the connection is established. The optimal sparse slot table is shown at the bottom right of Figure 1B.

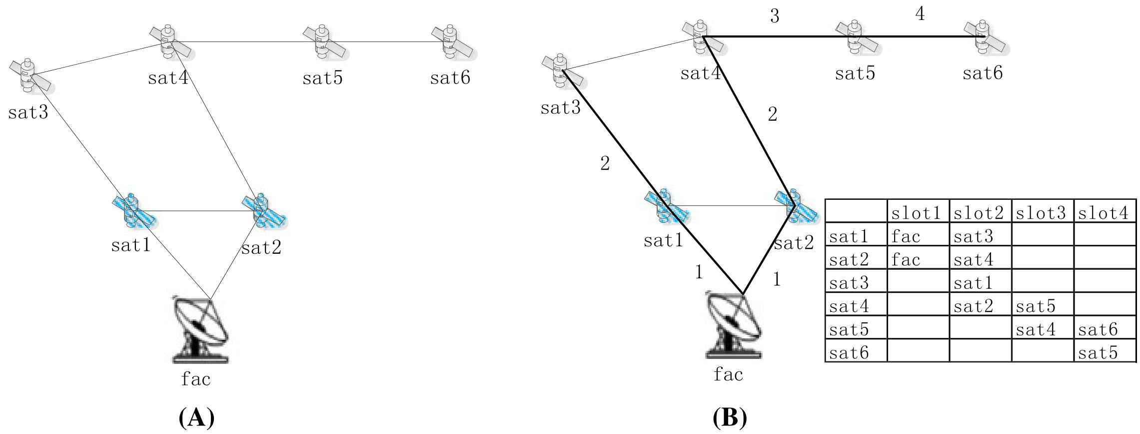
3.2 | Lower bound 2

Let T be the number of slots needed for the spread of data, n be the number of satellites, m be the number of source satellites, and q be the number of links that each satellite can maintain in one slot.

Proposition 2.

$$T \geq \left\lceil log_2 \frac{n}{m} + 1 \right\rceil \tag{11}$$

Proof: *If the adjacent matrix of the visibility relation is a complete graph (ie, all elements of the matrix equal 1 ($adj_{ij} = 1$, $\forall i, j$), then each satellite can create connections with any q satellites in any slot t, where q is the number of links that each satellite can maintain in one slot, which equals to 1. a_t is the total number of connections used during slot t. In the first slot, the stations send the data to all source*



	slot1	slot2	slot3	slot4
sat1	fac	sat3		
sat2	fac	sat4		
sat3		sat1		
sat4		sat2	sat5	
sat5			sat4	sat6
sat6				sat5

FIGURE 1 Diagram of the first lower bound. The figure illustrates lower bound 1, showing that the number of slots needed for the spread of data is greater than the maximal hop count. The visibility relationship is shown in A and the optimal topology shown in B [Colour figure can be viewed at wileyonlinelibrary.com]

satellites, and thus, $a_1 = m$. *In the following slots, each informed satellite can send the data to q different uninformed satellites, because the 2 satellites in any pair of satellites are visible to each other according to the hypothesis of the complete graph. Therefore,* $a_2 = qm$, $a_3 = q(m + qm) = qm(q + 1)$. *By mathematical induction, we then prove that* $a_t = qm(q + 1)^{t-2}$. *We already know that* $a_3 = qm(q + 1)^{3-2}$, *and we are thus ready to prove that* $a_t = qm(q + 1)^{t-2}$ *when* $t = k + 1$ *given that* $a_t = qm(q + 1)^{t-2}$, $t = 2, \ldots, k$ *and* $a_1 = m$:

$$a_t = q\sum_{t=1}^{t=t-1} a_t = q\left(m + qm + qm\,(q+1) + \cdots + qm\,(q+1)^{(t-1)-2}\right),$$

$$a_t = qm\left(1 + \frac{q\left(1-(q+1)^{t-2}\right)}{1-(q+1)}\right) = qm\left(1 + \left((q+1)^{t-2}\right)-1\right) = qm\,(q+1)^{t-2}.$$

The first 2 lines indicate that the number of connections established in slot t is the number of total informed satellites multiplied by q. To guarantee that all the satellites obtain the system data in T time slots, the summation of the number of informed satellites must be more than the number of satellites; ie,

$$\sum_{t=1}^{t=T} a_t \geq n.$$

We then have

$$\sum_{t=1}^{t=T} a_t = a_1 + \sum_{t=2}^{t=T} a_t = m + \sum_{t=2}^{t=T} qm\,(q+1)^{t-2} = m + qm\frac{1-(q+1)^{T-2}}{1-(q+1)},$$

$$m + qm\,\frac{1-(q+1)^{T-1}}{1-(q+1)} \geq n,$$

$$T \geq \log_{(q+1)}\frac{n}{m} + 1,$$

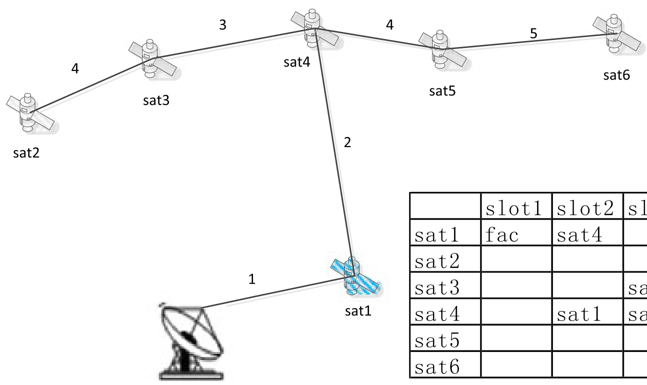
$$T \geq \left\lceil \log_{(q+1)}\frac{n}{m} + 1 \right\rceil,$$

where $\lceil\,\rceil$ denotes the rounding-up function. The last equation is obtained because T is a positive integer. When $q = 1$ as is assumed in this paper, $T \geq \left\lceil \log_2 \frac{n}{m} + 1 \right\rceil$. It should be noticed that q can take any positive integer value, which leads to a bound for a more general problem version than the variant addressed in the current paper.

3.3 | Analysis of the lower bounds

The lower bound of the objective function is the maximum value of the 2 kinds of lower bound; ie, LB = max(MHC, TCG), where LB is the lower bound of the objective function, MHC is the maximum hop count of all satellites from the source satellites, and TCG is the number of slots needed under the hypothesis of the complete graph.

The optimal solution of the problem will be greater than the lower bound if there is a satellite that needs to transmit data for more than one branch assuming that $q = 1$. For example, in Figure 2, sat1 is the only source satellite, and sat2 and sat6 have the same hop counts of 4 as the maximal hop counts of the network. Sat4 cannot send the data to sat3 and sat5 in the same slot because each satellite only establishes one connection in a single slot. Therefore, the optimal solution, which is 5, needs 1 more slot for data spread than the lower bound of the problem. The optimal sparse slot table is presented at the bottom right of the figure. This demonstrates that the optimal solution may be greater than the lower bound if there is a satellite that is responsible for transmitting the data in more than one branch.



	slot1	slot2	slot3	slot4	slot5
sat1	fac	sat4			
sat2				sat3	
sat3			sat4	sat2	
sat4		sat1	sat3	sat5	
sat5				sat4	sat6
sat6					sat5

FIGURE 2 Case where the lower bound is not the optimal solution. The figure shows the situation that both lower bounds are less than the optimal solution. This is because sat4 is responsible for transmitting the data in more than one branch [Colour figure can be viewed at wileyonlinelibrary.com]

4 | DETERMINISTIC CONSTRUCTIVE ALGORITHM FOR THE TDILTG PROBLEM

This section designs an algorithm that solves the problem effectively. From the previous section, we know that it is difficult to compute the optimal solution of the problem quickly. According to the practical needs of engineering, we design a heuristic algorithm that solves the TDILTG problem effectively. The algorithm focuses on constructing a good sequence of uninformed satellites and arranging the uninformed satellites such that they link to an appropriate informed satellite one by one. The algorithm gives priority to satellites that have fewer opportunities to receive the data, and it is called the deterministic constructive algorithm or DC algorithm for short.

4.1 | Notations of the DC algorithm

The main idea of the DC algorithm is to successively select a destination satellite, whose number of visible satellites is minimal but not zero, and arrange it such that an appropriate informed satellite provides the destination satellite the earliest receiving slot until all the satellites receive the data. When there is a tie between satellites that can provide the earliest receiving of data, we can break it by selecting the satellite whose number of visible satellites is lower.

To explain the algorithm below, we introduce new notations used in the algorithm here.

1. C_i and x_{ijt} are the same as mentioned above;
2. S is the set of source satellites and $|S| = m$ states that the number of the source satellites is m;
3. O is the set of destination satellites that are uninformed in the initialization of the algorithm. $|O| = k$ means that the number of destination satellites is k. Consequently, m plus k equals n, which is the total number of satellites;
4. A is the set of informed satellites;
5. U is the set of uninformed satellites;
6. VO_i is the set of uninformed satellites that are visible to satellite i, and $|VO_i|$ is the number of elements in VO_i; and
7. VI_i is the set of informed satellites that are visible to satellite i, and $|VI_i|$ is the number of elements in VI_i.

4.2 | Details of the DC algorithm

The steps of the algorithm are as follows, where the capital letters mainly denote the collection variables and the lowercase letters mainly denote the numerical variables.

- Step 1 is to initialize the algorithm, arrange the stations, establish connections with the source satellites in the first slot, add the source satellite set S to the informed satellite set A and set the receiving slots of source satellites to 1, and add the destination satellite set O to the uninformed satellite set U and set the receiving slots of these satellites to ∞.
- Step 2 is to select an uninformed satellite whose number of visible informed satellites is minimal but not zero as satellite s, while a tie is broken by selecting the satellite whose number of visible uninformed satellites is maximal.
- Step 3 is to set the current slot t to 2.
- Step 4 is to judge whether satellite s can establish a connection with an informed satellite in slot t, and to then set $t = t + 1$ and to return to the beginning of Step 4 if there are no such satellites, or to select the satellite q whose number of visible satellites is smallest otherwise.
- Step 5 is to arrange satellite s such that it establishes a connection with satellite q, and then to set $C_i = t$.
- Step 6 is to update all the parameters, delete satellite s from set U and to add it to set A, and to update the sets VO and VI of all the satellites.
- Step 7 is to judge whether set U is empty, return to Step 2 to select the next destination satellite if it is not, and to go to Step 8 otherwise.
- Step 8 is to organize the receiving slot and corresponding sending-data satellites of each satellite, and to generate the topology of the inter-satellite link network.

The pseudocode of the DC algorithm is given in Figure 3.

4.3 | Time complexity of the DC algorithm

We analyze the time complexity of the DC algorithm, which is $O(n^3)$. The algorithm selects a destination satellite one by one and then arranges it in an appropriate slot. The number of destination satellites is less than n, which is the total number of satellites, ie, $|O| < n$. For each selected destination satellite, the maximal t is also less than n because the visibility relation matrix is connected, which means that at least 1 uninformed satellite will receive the data in a new slot. In each slot, it is obvious that the operation to select satellite q is a linear function whose time complexity is $O(n)$. The complexity of the DC algorithm is therefore $O(n^3)$. This indicates that the algorithm can deal with large-scale problems and solve the TDILTG problem rapidly.

```
function DC-algorithm (S, O, adj) returns a solution
inputs: S, the set of source satellites
        O, the set of destination satellites
        adj, the visibility relation matrix of satellites
//initialization
A←S; U←O;
for each i in S do
    C_i = 1;
for each j in O do
C_j = ∞;
for each i in S and O do
    VO_i ← the number of visible uninformed satellites of satellite i;
    VI_i ← the number of visible informed satellites of satellite i;
for all thei,j,t do
x_ijt ← 0;
//determining whether to end
while U ≠ ∅
//selection the destination satellite
    S ← {s| argmin_s |VI_s| , |VI_s| > 0, s ∈ U}
    if |S|>1
        s ← {s′| argmax_s′ |VO_s′| , s′ ∈ I}
    else
        s ← {s|s ∈ S}
    //arranging receiving slot
   t ← 2;
   q ← {k| argmin_k |VO_k| , |VO_k| > 0, k ∈ A, k ∈ VI_s, x_kjt = 0, ∀j}
   while q=∅
        t ← t+1;
        q ← {k| argmin_k |VO_k| , |VO_k| > 0, k ∈ A, k ∈ VI_s, x_kjt = 0, ∀j}
   C_s ← t;
   x_sqt ← 1;
   //updating the parameters
   A ← A ∪ {s};
   U ← U / {s};
   foreach i in S and O do
   update(VO_i);
   update(VI_i);
```

FIGURE 3 Pseudocode of the deterministic constructive algorithm

5 | EXPERIMENT

In this section, we first show how we generate the test instances according to the demands of engineering. We then validate the DC algorithm and analyze the effects of different instance parameters through numerical experiments.

5.1 | Test instances

According to the analysis of the TDILTG problem, there are 3 input parameters of the model, namely, the number of source satellites, the number of total satellites, and the matrix of the visibility relationship. To test the DC algorithm systematically, we select a number of values of the first 2 parameters and generate the matrix of the visibility relationship via a random graph. An edge for any pair of nodes occurs with probability P, which means any 2 satellites are visible to each other with probability P. We use the tuple {N, P, S} to represent a combination of parameters for an example, where N is the number of satellites, P is the probability of visibility between the satellites of any satellite pair, and S is the number of source satellites. Given a tuple of parameters, we first generate a random graph of N nodes with probability P to build edges, and then randomly select S satellites as the source satellites.

For practical application, we should make the visible matrix connected, which means there is a proper transmission path between any 2 given satellites. And then we divide the instances into 2 parts, which are the small-scale instances and the large-scale instances. The small-scale instances are designed to verify that the DC algorithm can meet the practical demand in engineering, when the number of satellites is close to the actual number of the global satellite navigation system. According to the demand of engineering, all the satellites should receive the data before the end of the horizon, where the T_{max} is 15, and the computation time of the time-division topology should be less than 2 seconds. The large-scale instances are designed to verify that the DC algorithm is also effective when the number of satellites is large enough, which means the DC algorithm has a strong applicability.

Generally, the total number of satellites in a global satellite navigation system ranges between 20 and 30, the number of satellites visible to each satellite ranges between 3 and 6, and the ground stations can use 1 to 4 source satellites to spread data to all satellites. The domain of the 3 parameters of the small-scale instances is thus presented in Table 1. And the parameter q is 1, which is the number of links that each satellite can maintain in one slot. There are in total 484 (11 × 11 × 4 = 484) tuples of parameters. Given any tuple, we generate 1000 random graphs to validate the algorithm.

The major difference between the large-scale instances and the small-scale instances is the number of satellites. In the large-scale instances, the number of satellites ranges from 20 to 40 to analyze the performance of the DC algorithm as the problem size gradually increases. On the other hand, we reduce the value of parameter P, which is the probability of visibility between the satellites of any satellite pair, to further increase the difficulty of solving the problem. As for the number of source satellites, we keep the domain of S equal to the ones of the small-scale instances. The settings for parameter q is kept unchanged as well, i.e., q = 1. The domain of the 3 parameters of the large-scale instances is thus presented in Table 2. There are in total 440 (10 × 11 × 4 = 440) tuples of parameters. Given any tuple, we generate 100 random graphs to validate the algorithm.

5.2 | Test results of the small-scale instances

To evaluate the performance of the proposed DC algorithm, we collect statistics of the nonlower-bound (NLB) solutions and the average number of slots used to spread data. A lower percentage of NLB solutions mean that the DC algorithm has a greater probability of obtaining the optimal solution for a certain combination of parameters; however, an NLB solution may also be an optimal solution as shown in Figure 2. The average number of slots for data spread is the average number of slots needed to spread data to all satellites in 1000 instances for a certain combination of the parameters; this value shows the effects of different parameters on the required number of slots for data spread when using the DC algorithm.

The percentage and number of NLB solutions given different numbers of source satellites are presented in Table 3. There are 121 000 examples for each number of source satellites. When there are 4 source satellites, the largest number of NLB solutions is 15 963, which is accounting for 13.19% of the total number of cases. On average for all numbers of source satellites, NLB solutions account for 8.07% of all cases. This means the algorithm obtains the optimal solution in at least 91.93% of all the cases. In the NLB solutions, the difference between the solutions obtained by the DC algorithm and the lower bounds is 1 slot in most cases, and the maximal slot difference is 2 slots. We used 484 000 instances to test our DC

TABLE 1 Domain of the 3 parameters of small-scale instances

Parameter	Value										
N	20	21	22	23	24	25	26	27	28	29	30
P	0.10	0.11	0.12	0.13	0.14	0.15	0.16	0.17	0.18	0.19	0.20
S	1	2	3	4							

TABLE 2 Domain of the 3 parameters of large-scale instances

Parameter	Value										
N	20	40	60	80	100	120	140	160	180	200	
P	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.14	0.15
S	1	2	3	4							

TABLE 3 Statistics of nonlower-bound (NLB) solutions in the small-scale instances

Number of Source Satellites	Number of NLB Solutions	The Percent of NLB Solutions
1	3908	3.23%
2	8381	6.93%
3	10 789	8.92%
4	15 963	13.19%
Total	39 041	8.07%

algorithm in MATLAB 2014a on a computer with a 2.4-GHz i5CPU and 3.79-GB RAM, and the total time taken was 32 948.5 second. A single instance therefore takes about 0.68 seconds, which meets the demands of engineering well.

Figure 4 shows the number of NLB solutions for different tuples of the parameters. It is seen that, for some combinations of parameters, it is difficult to get the lower bound using the DC algorithm, and the variation tendency of the percentage of NLB solutions is different for different numbers of source satellites. Further analysis was performed for these findings.

Figure 5A shows the effect of different values of N on the NLB solutions of small-scale instances, which gets from the addition of different values of parameter P under each combination of N and S. When there is one source satellite, there are more NLB solutions for a larger value of N. When there are 2 source satellites, the parameter N has little effect on the number of NLB solutions. When there are 3 source satellites, the number of NLB solutions increases with N below 24 and then decreases sharply for N greater than 25. This is because the expression

$$\left\lceil \log_2 \frac{N}{m} + 1 \right\rceil - \left(\log_2 \frac{N}{m} + 1 \right) \tag{12}$$

is zero when N is 24, and decreases first and then increases as N increases between 20 and 30.

According to the DC algorithm, it is difficult to get the lower bound when the result of Equation 12 is small; we will design a special algorithm for such cases in future work. For example, there are few NLB solutions when N is greater than 25, owing to the increment of the value of Expression 12. When there are 4 source satellites, the number of NLB solutions increases sharply as N increases from 23 to 24, and then decreases slowly because the maximal hop count (lower bound 1) of the network is greater when there are more satellites.

Figure 5B shows the effect of different values of P on the NLB solutions of small-scale instances, which gets from the addition of different values of parameter N under each combination of P and S. It is seen that there are fewer NLB solutions for a larger probability P. This is because a large value of P provides more transmission paths among satellites, making it less possible for a satellite to be responsible for transmitting data in more than one branch. Figure 5B also shows that there are more NLB solutions if there are more source satellites. A greater number of source satellites increase the complexity of the problem and decrease the lower bound of the problem, and it is thus more difficult to get to the lower bound solutions of the problem when the scale of the problem is small.

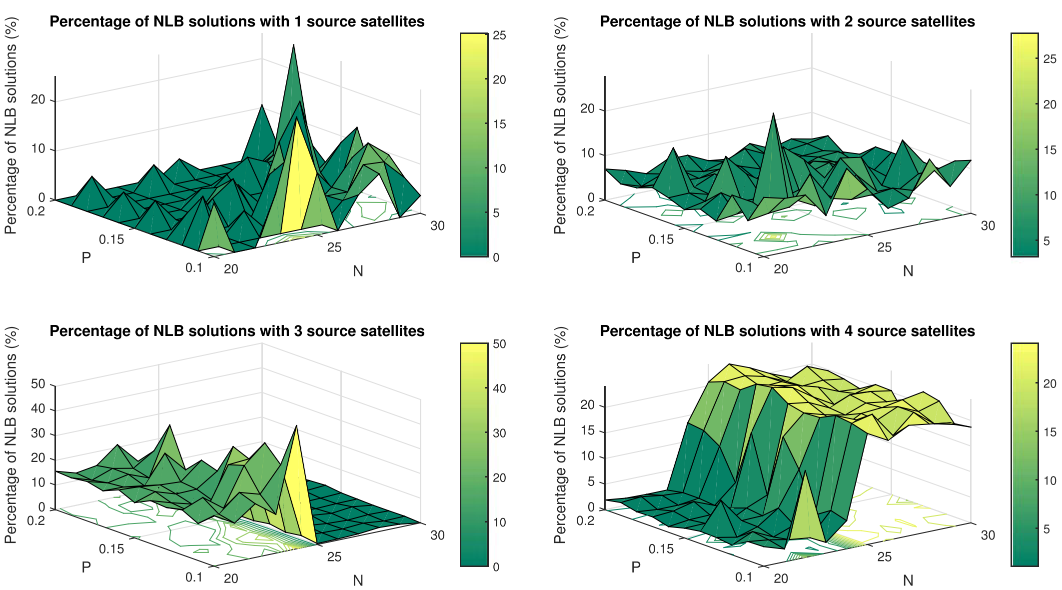


FIGURE 4 Percentage of nonlower-bound (NLB) solutions for different parameters in small-scale instances. This figure presents the variation tendencies of NLB solutions given different numbers of source satellites. In each subgraph, the percentage of NLB solutions is shown given a combination of N and P, which are the number of nodes and the probability of building edges respectively [Colour figure can be viewed at wileyonlinelibrary.com]

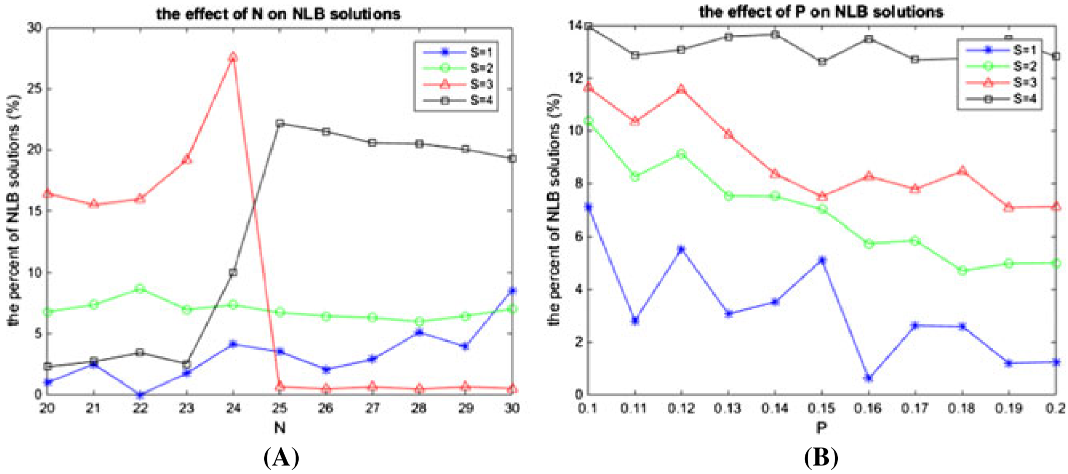


FIGURE 5 Effect of parameters N and P on the nonlower-bound (NLB) solutions in small-scale instances. A, The effect of parameter N on the percentage of NLB solutions given different numbers of source satellites. B, The effect of parameter P on the percentage of NLB solutions given different numbers of source satellites [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 6 shows the average number of slots used to spread data for different tuples of the parameters. When there is 1, 2, or 4 source satellites, the changing trends of the average number of slots are almost the same as those of the number of NLB solutions. The first reason is that, to spread data, an NLB solution needs one more slot than the lower bound of the problem in most cases. The second reason is that the lower bound of the problem remains basically stable. When there are 3 source satellites, the average number of slots jumps to 5 when N increases from 24 to 25 owing to the abrupt change in the lower bound of the problem as stated earlier.

Figure 7A shows the effect of N on the average number of slots, which gets from the addition of different values of parameter P under each combination of N and S. The average number of slots decreases as the number of source satellites increases. The effect of N is weak when there is 1 or 2 source satellites. When there are 3 source satellites, the abrupt change in the lower bound results in the average number of slots jumping to 5. When there are 4 source satellites, the average number increases when N increases from 23 to 25, and then decreases slowly. The average number of slots for 3 source satellites is a little larger than that for 4 source satellites when N is less than 24. Furthermore, the average number of slots for 3 source satellites is almost the same as that for 2 source satellites when N is greater than 25. These 2 findings can guide the selection of the number of source satellites in engineering application.

P has very little effect on the average number of slots given different numbers of source satellites, as shown in Figure 7B, which means the average number of satellites visible to each satellite has little effect on the average number of slots; this is because the slots needed are very close to the lower bound of the problem given any parameter P.

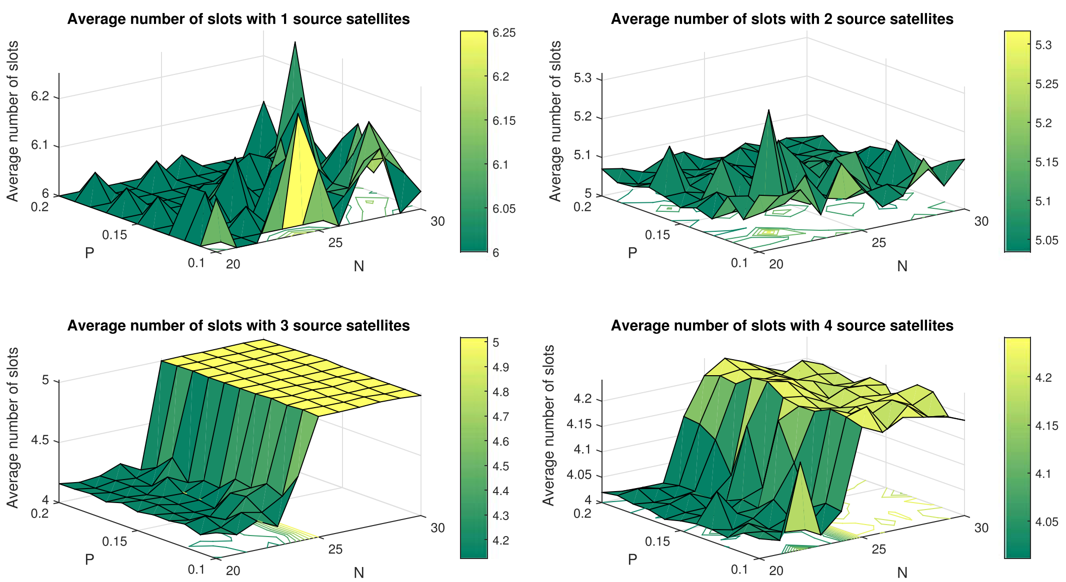


FIGURE 6 Average number of slots for different parameters of small-scale instances. The figure presents the variation tendencies of the average number of slots needed to spread data obtained by the DC algorithm given different numbers of source satellites. In each subgraph, the average number of slots is shown given a combination of N and P, which are the number of nodes and the probability of building edges respectively [Colour figure can be viewed at wileyonlinelibrary.com]

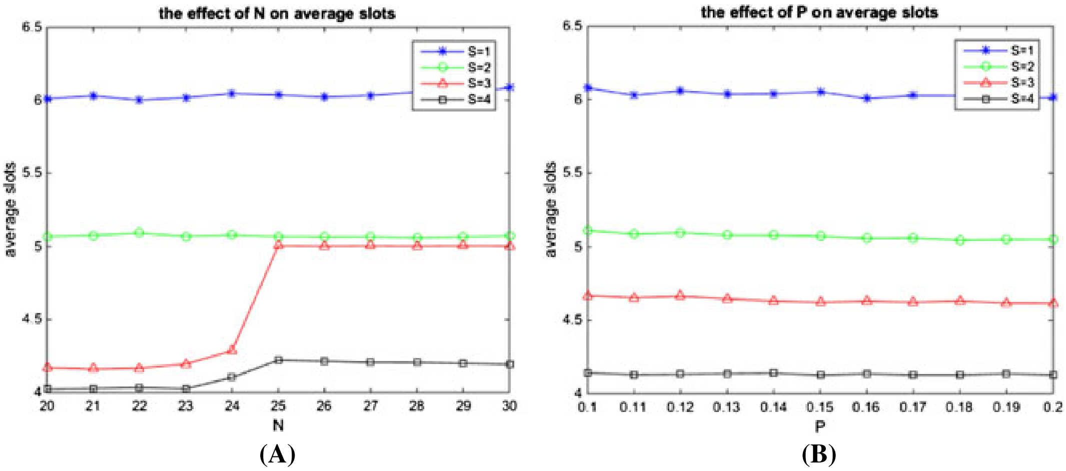


FIGURE 7 Effect of parameters N and P on the average number of slots of small-scale instances. A, Effect of parameter N on the average number of slots needed to spread data obtained by the DC algorithm. B, Effect of parameter P on the average number of slots needed to spread data obtained by the DC algorithm [Colour figure can be viewed at wileyonlinelibrary.com]

5.3 | Test results of the large-scale instances

In the large-scale instances, we collect the statistics of the NLB solutions but not the average number of slots used to spread data. This is because we only aim to validate the performance of the algorithm in large-scale instances, and the average number could be easily estimated by the lower bounds in the previous statement if there is a low percentage of NLB solutions. As is discussed in the last chapter, a lower percentage of NLB solutions mean that the DC algorithm has a greater probability of obtaining the optimal solution for a certain combination of parameters; however, an NLB solution may also be an optimal solution as shown in Figure 2.

The percentage and number of NLB solutions given different numbers of source satellites are presented in Table 4. There are 11 000 examples for each number of source satellites. When there are 3 source satellites, the largest number of NLB solutions is 113, which is accounting for 1.03% of the total number of cases. On average for all numbers of source satellites, NLB solutions account for 0.86% of all cases. This means the algorithm obtains the optimal solution in at least 99.14% of all the large-scale cases. In the NLB solutions, the difference between the solutions obtained by the DC algorithm and the lower bounds is 1 slot in most cases, and the maximal slot difference is 2 slots. We used 44 000 instances to test our DC algorithm in MATLAB 2014a on a computer with a 2.4-GHz i5CPU and 3.79-GB RAM, and the total time taken was 42 240.6 seconds. A single instance therefore takes about 0.96 second, which meets the demands of engineering well.

Figure 8 shows the number of NLB solutions for different tuples of the parameters. It is seen that there is the lower percentages of the NLB solution as the number of satellites increases. This means the DC algorithm is still effective when the scale of the problem increases, which demonstrates the algorithm could solve the TDILTG problem efficiently.

Figure 9A shows the effect of different values of N on the NLB solutions of large-scale instances, which gets from the addition of different values of parameter P under each combination of N and S. The results indicate that it is more difficult to get the lower bound when the number of satellites is smaller, so the DC algorithm may be more suitable for large-scale problems. However, the algorithm still achieves satisfactory when the scale of the problem is small, proved by the highest percentage of NLB solutions is less than 7%. The figure verifies the conclusion that it is difficult to get the lower bound when the result of Equation 12 is small. For example, the percentage of NLB solutions is relatively high when there are 60 or 120 satellites with 1 or 2 source satellites.

Figure 9B shows the effect of different values of P on the NLB solutions of large-scale instances, which gets from the addition of different values of parameter N under each combination of P and S. It is seen that there are fewer NLB solutions for a larger probability P. This is because a large value of P provides more transmission paths among satellites, making it less possible for a satellite to be responsible for transmitting data in more than one branch.

TABLE 4 Statistics of nonlower-bound (NLB) solutions in the large-scale instances

Number of Source Satellites	Number of NLB Solutions	The Percent of NLB Solutions
1	134	1.22%
2	111	1.01%
3	113	1.03%
4	19	0.17%
Total	377	0.86%

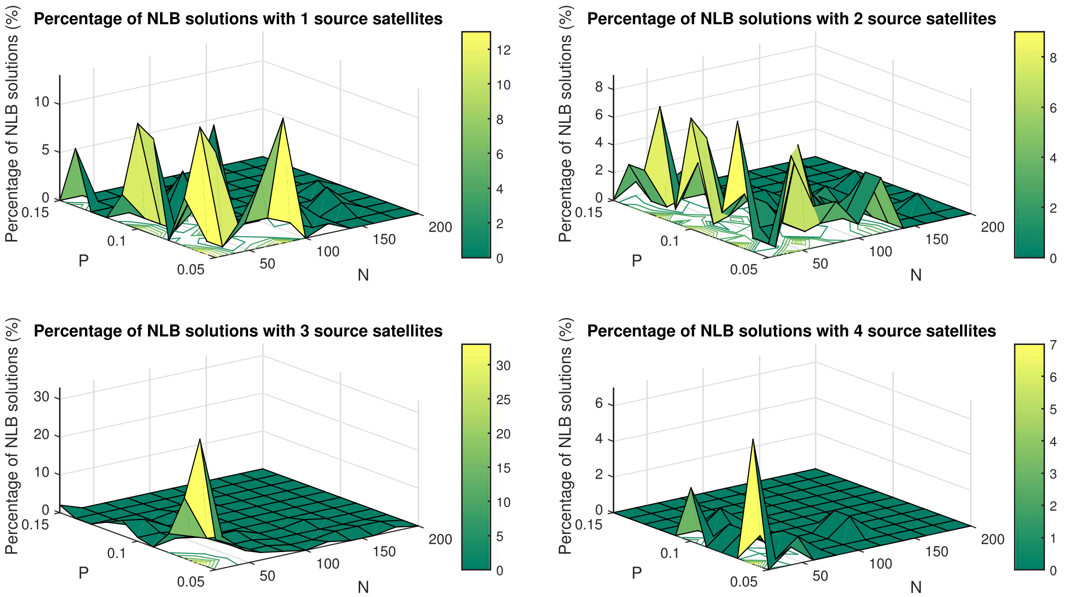


FIGURE 8 Percentage of nonlower-bound (NLB) solutions for different parameters of large-scale instances. This figure presents the variation tendencies of NLB solutions given different numbers of source satellites. In each subgraph, the percentage of NLB solutions is shown given a combination of N and P, which are the number of nodes and the probability of building edges respectively [Colour figure can be viewed at wileyonlinelibrary.com]

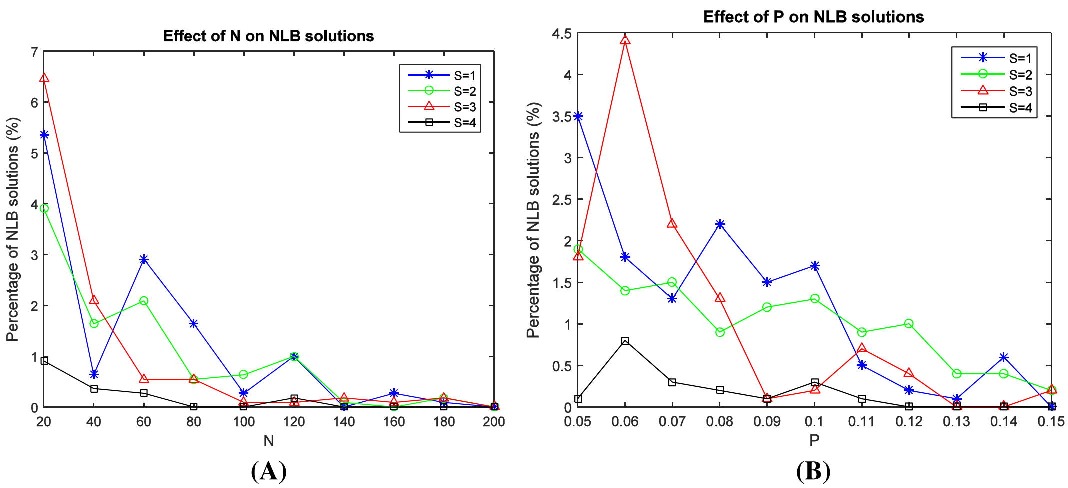


FIGURE 9 Effect of parameters N and P on the nonlower-bound (NLB) solutions of large-scale instances. A, The effect of parameter N on the percentage of NLB solutions given different numbers of source satellites. B, The effect of parameter P on the percentage of NLB solutions given different numbers of source satellites [Colour figure can be viewed at wileyonlinelibrary.com]

6 | CONCLUSION

This paper studies the TDILTG problem. Being the first attempt to investigate the TDILTG problem, we proposed a mathematical model to formally define the problem and obtained 2 lower bounds through a careful analysis of the problem properties. We also proposed a very effective deterministic constructive algorithm to solve the problem approximately (which gives an upper bound of the problem). Experimental results on a wide range of instances validate the effectiveness of the proposed DC algorithm. In particular, the DC algorithm is able to solve the problem to optimality (where the upper bound matches the lower bound) in most cases with a computational time less than 1 second. For other cases where optimality is not proved, the upper bound is very close to the lower bound with a gap of only 1 or 2 units. Finally, we also analyzed the effects of different instance parameters on the number of slots needed for the spread of data, which provides guidelines for engineering application.

In future work, we will study how the structures of the sources affect the spread of data in an inter-satellite link network. We will then generate a topology of the inter-satellite link network with consideration of other the inter-satellite crosslink demand.

ACKNOWLEDGEMENT

We are grateful to the reviewers for their insightful comments, which helped us to improve the paper. This work was supported by the National Natural Science Foundation of China (Grant number 71201171).

REFERENCES

1. Wang YP, Zhang G, Jiang ZQ. A novel routing algorithm design of time evolving graph based on pairing heap for MEO satellite network. 2014 Ieee 80th vehicular technology conference, Vancouver. Sep 14–17, 2014.
2. Yong-ping X, Li-min S. Opportunistic networks. *J Softw*. 2009;20(1):124-137.
3. Dorogovtsev SN, Mendes JFF. *Evolution of Networks*. Oxford, NY: Oxford University Press; 2003.
4. Clementi A, Pasquale F, Silvestri R. Opportunistic manets: mobility can make up for low transmission power. *IEEE/ACM Trans Networking*. 2013;21(2):610-620.
5. Ros FJ, Ruiz PM. Minimum broadcasting structure for optimal data dissemination in vehicular networks. *IEEE Trans Veh Technol*. 2013;62(8):3964-3973.
6. Eiza MH, Ni Q. An evolving graph-based reliable routing scheme for VANETs. *IEEE Trans Veh Technol*. 2013;6(24):1493-1504.
7. Rossi R A, Gallagher B, Neville J, et al.: Modeling dynamic behavior in large evolving graphs. In Proceedings of the 6th ACM international conference on Web search and data mining. Rome, Italy; 2013:667–676
8. Shu J, Jiang S, Liu Q, Liu L, Geng X. A connectivity monitoring model of opportunistic sensor network based on evolving graph. *Comput Sci Inf Syst*. 2015;1(12):895-909.
9. Abdalla GMT, Abu-Rgheff MA, Senouci SM. IEEE Global Inf. Infrastruct. SympCurrent trends in vehicular ad hoc networks. In Proc., Marrakech, Morocco; 2007:1–9.
10. Blum JJ, Eskandarian A, Hoffman LJ. Challenges of inter vehicle ad hoc networks. *IEEE Trans Intell Transp Syst*. 2004;5(4):347-351.
11. Mao G, Anderson BDO. Graph theoretic models and tools for the analysis of dynamic wireless multihop networks. In Proc. IEEE Wireless Commun. Netw. Conf., 2009:1–6.
12. Monteiro J, Goldman A, Ferreira A. Performance evaluation of dynamic networks using an evolving graph combinatorial model. In Proc.IEEE Int. Conf. WiMob Comput., Netw. Commun., 2006:173–180.
13. Bhadra S, Ferreira A. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. 2nd International Conference on Ad-Hoc, Mobile and Wireless Networks. Montreal, October 08–10, 2003.
14. Avin C, Koucky M, Lotker, Z. How to explore a fast-changing world (cover time of a simple random walk on evolving graphs). 35th International Colloquium on Automata, Languages and Programming. Keykjavik, July 07–11, 2008.
15. Chan J, Bailey J, Leckie C. Discovering correlated spatio-temporal changes in evolving graphs. *Knowl Inf Syst*. 2008;16:53-96.
16. Clementi AEF, Monti A, Pasquale F, et al. Information spreading in stationary Markovian evolving graphs[J]. *IEEE Trans Parallel Distrib Syst*. 2011;9(22):1425-1432.
17. Grindrod, Peter, Higham, Desmond J. Evolving graphs: dynamical models, inverse problems and propagation. Proceedings Of the Royal Society a-Mathematical Physical And Engineering Sciences, 2010
18. Clementi AEF, Macci C, Monti A, et al. Flooding time of edge-markovian evolving graphs [J]. *SIAM J Discret Math*. 2010;34:1694-1712.

Xiaogeng Chu was born in Liaoyang, China, in 1990. He received the Bachelor degree in 2011 from Shanghai Jiao Tong University, and received the master degree in management science and engineering in 2013 from National University of Defense Technology (NUDT). He is currently a PHD candidate at NUDT. His research interest covers optimization methods and their applications to the modeling, navigation satellite communication optimization problem and optimization scheduling of earth observation satellites.

Yuning Chen received the B.Eng. and M.Eng. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 2010 and 2012 respectively, and received the Ph.D. degree in Computer Science at the LERIA laboratory, University of Angers, France, in 2016. He is currently an assistant professor at NUDT. His research interests include developing efficient algorithms for solving photograph satellite scheduling problem, navigation satellite communication optimization problem and other classical combinatorial optimization problems.

How to cite this article: Chu X, Chen Y. Time division inter-satellite link topology generation problem: Modeling and solution. *Int J Satell Commun Network*. 2017;0:1-13. https://doi.org/10.1002/sat.1212