

运用动态规划的智能卫星集群星间通信路由算法*

伦伟成¹, 李群¹, 于芹章², 张灿¹

(1. 国防科技大学 系统工程学院, 湖南长沙 410073; 2. 复杂系统仿真总体重点实验室, 北京 100101)

摘要:通过建立智能卫星集群网络模型,把智能卫星集群星间通信路由问题转换为时延最短路径问题,进而提出一种求解此问题的智能卫星集群星间通信路由算法。该路由算法采用动态规划策略分阶段规划智能卫星集群两个成员之间的星间通信路由,在每个规划阶段,负责发送数据的智能卫星自主调用一种星间通信路由由静态规划算法,以求出其在当前时刻的后继卫星来转发数据。该算法还能够处理星间链路传输能力有限而导致数据需要分段传送的问题。使用基于 Agent 的建模与仿真方法进行了大量仿真实验,从而考察并分析了不同智能卫星集群使用前述路由算法规划星间通信路由的效果。

关键词:智能卫星集群;星间通信路由;动态规划;智能体

中图分类号:TP391.9 文献标志码:A 文章编号:1001-2486(2022)01-137-09

Inter-satellite communication routing algorithm of smart-satellites-swarm using dynamic programming

LUN Weicheng¹, LI Qun¹, YU Qin Zhang², ZHANG Can¹

(1. College of Systems Engineering, National University of Defense Technology, Changsha 410073, China;

2. Science and Technology on Complex Systems Simulation Laboratory, Beijing 100101, China)

Abstract: The network of SSS (smart-satellites-swarm) was modelled in order to transfer the ICRPS (inter-satellite communication routing problem of SSS) into the least-delay path problem. And the ICRAS was developed to solve this problem. ICRAS used dynamic programming method to program the inter-satellite communication routing between two members of a SSS in multiple phases. The smart satellite which need transmit data adopted a static programming algorithm for inter-satellite communication routing to find its successor satellite during each programming phase. ICRAS was able to cope with the problem that data had to be delivered in pieces since the propagation capability of an inter-satellite link was restricted. A large quantity of simulation experiments were designed and conducted to examine and analyze the performances of ICRAS when different SSS utilized ICRAS to program inter-satellite communication routings.

Keywords: smart-satellites-swarm; inter-satellite communication routing; dynamic programming; agent

智能卫星是具有自主能力的一类人造卫星,它能够在尽可能减少地面指令的直接操控下独立完成任务,如此可降低操作成本且节省通信资源。单个智能卫星的能力有限,将多个智能卫星编组为智能卫星集群(Smart-Satellites-Swarm, SSS)则可以增强卫星的多任务遂行能力和抗干扰能力,提高卫星完成复杂航天任务的效率和成功率。以色列的 SAMSON 计划^[1]和美国的小行星探测计划^[2]都是典型的智能卫星集群实例。

要促进从智能卫星的“单体自主”向 SSS 的“群体智能”发展,必须使卫星具备网络化信息协同能力,在此基础上建立有效的星间信息协同机制,让 SSS 的成员卫星可以借助星间链路进行通信

以实现信息交互和任务协作,从而更好地完成那些对时效性、响应性、自主性都有较高要求的任务。为了达到前述目标,亟待解决智能卫星集群的星间通信路由问题(Inter-satellite Communication Routing Problem of Smart-satellites-swarm, ICRPS)。ICRPS 的核心是如何规划数据从源卫星传到目的卫星的合理路径,使星间通信在时延等方面满足任务要求。

星间通信路由问题可以在卫星网络的背景下进行研究,卫星网络的节点是卫星,边是星间链路。Xu 等^[3]提出了基于编码的多路径路由算法以应对多层卫星网络中的通信拥塞问题。Xie 等^[4]面向动态星间网络提出了采用广义领域搜

* 收稿日期:2020-11-14

基金项目:国家自然科学基金资助项目(62003359)

作者简介:伦伟成(1994—),男,湖南长沙人,博士研究生,E-mail:lunweicheng@qq.com;

李群(通信作者),男,教授,博士,博士生导师,E-mail:liqun@nudt.edu.cn

索的星间路由算法。Huang 等^[5]采用梯级优化设计方法来设计导航卫星网络的星间链路联络方案。Wang 等^[6]面向低轨卫星通信网络提出了基于软件定义网络架构的改进路由算法。尤启迪等^[7]针对面向卫星网络的空间信息快速回传场景提出了多路径快照聚合路由算法。Soret 等^[8]面向低轨元星座卫星网络提出了通过最小化轨道面间链路来自动分配路由的算法。Zhang 等^[9]面向低轨卫星网络提出了基于星间链路状态信息的星上自动控制的路由算法。Roth 等^[10]针对双低轨 Walker 星型星座设计了采用星间链路的两层地理路由体制。梁俊等^[11]针对软件定义卫星网络提出了基于切比雪夫神经网络的智能路由策略。

上述算法主要针对传统卫星,在工作模式方面还是由地面的指控中枢预先规划路由,且路由一旦确定便不再更改,卫星在其中只能机械地收发数据,若将这些算法用于 ICRPS,则无法发挥智能卫星的自主性优势。

因此本文研究了一种适用于智能卫星的星间通信路由算法,该算法基于动态规划的思想,由各卫星自主参与规划路由,使其在规划时能够根据 SSS 现状灵活调整路由,达到路由在数据传送过程中持续动态更新的效果。

1 ICRPS 概述

为了求解 ICRPS,需要基于 SSS 建立智能卫星集群网络(Network of Smart-Satellites-Swarm, NSSS):将 SSS 各成员抽象为节点,成员之间存在的星间链路抽象为边,边的权值就是星间链路的长度;从而把 ICRPS 转换为求解 NSSS 的两个节点间的最优路径。

不过,ICRPS 寻求的最优路径要满足时延最短而非长度最短。时延是指数据从源卫星传送到目的卫星所需的时间,ICRPS 主要考虑以下两种时延:

1) 发送时延,指从发送数据的第一个比特算起到其最后一个比特发送完毕所需的时间^[12],等于数据长度与发送速率之商;

2) 传播时延^[12],指电信号传播星间链路长度的距离所需的时间,等于星间链路长度与电信号传播速率(约等于光速)之商。

因 NSSS 的节点处在不断运动中,节点间时延和链路连通性也在不断发生变化^[11],故 NSSS 其实属于动态网络;但在某个具体时刻 NSSS 又是静态网络,因为卫星在此时刻的位置是固定的。考虑到 NSSS 的这种二元特性,结合智能卫星的自主性,采用动态规划策略求解 ICRPS,分多个阶

段规划 SSS 成员星间通信的路由,在每一阶段中把 NSSS 作静态网络处理,各阶段的 NSSS 整合起来又是随时间变化的动态网络。

采用图论的表示方法^[13],将 NSSS 记作 (V, A) , V 表示节点集(即卫星集合), A 表示边集(即星间链路集合),节点数记为 $|V| = n$,再构造 NSSS 的邻接矩阵 $Q = (q_{ij})_{n \times n}$, q_{ij} 的赋值规则如式(1)所示。

$$q_{ij} = \begin{cases} l_{ij} & v_i \text{ 与 } v_j \text{ 有长度为 } l_{ij} \text{ 的星间链路} \\ \infty & v_i \text{ 与 } v_j \text{ 没有星间链路} \end{cases} \quad (1)$$

为便于表达,规定:卫星 v_i 、节点 v_i 和 v_i 这三种表达是等价的;边和星间链路含义相通;边权值和星间链路长度含义相通。

包含 v_i 和 v_j 的卫星对能否建立星间链路(简称“建链”)取决于是否同时满足以下两个条件:

1) 卫星对彼此可见,要求地心到卫星对连线的直线距离 h_L 大于地球强电离层高度 h_1 与地球半径 R_E 之和,即

$$h_L > h_1 + R_E \quad (2)$$

2) 卫星对可以互相通信,要求 v_i 对于 v_j 的俯仰角 θ_i 在 v_i 的天线扫描范围之内且 v_j 对于 v_i 的俯仰角 θ_j 在 v_j 的天线扫描范围之内,即

$$\begin{cases} \gamma_{i,\min} < \theta_i < \gamma_{i,\max} \\ \gamma_{j,\min} < \theta_j < \gamma_{j,\max} \end{cases} \quad (3)$$

式中, $\gamma_{i,\min}$ 和 $\gamma_{j,\min}$ 分别表示 v_i 和 v_j 的天线扫描范围的下限, $\gamma_{i,\max}$ 和 $\gamma_{j,\max}$ 表示上限。

卫星对若能建链,便可以进行通信,此时其星间链路的长度就等于卫星之间的直线距离。

作为智能卫星,SSS 各成员可以采用文献[14]中的方法根据轨道根数计算其他成员在惯性坐标系的位置,进而在考察是否满足式(2)和式(3)所示条件的基础上得到当前时刻的 NSSS。在必要时,SSS 将通过地面站上注星历等方式保证计算的准确性。

2 星间通信路由静态规划算法

2.1 算法描述

星间通信路由静态规划算法借鉴了 Dijkstra 算法^[15]的思想,用于求解某一具体时刻的静态 NSSS 上一条以 v_b 为起点、 v_e 为终点的时延最短路径。

设 $v_i, v_j \in V$, 定义有向链表 $P_j(v_i)$ 表示从 v_j 到 v_i 的路径(未必最优), $P_j(v_i)$ 的长度记作 $L_j(v_i)$, $P_j(v_i)$ 的节点数记作 $|P_j(v_i)|$, 从而构造函数 $T_j(v_i)$ 描述数据传输的总时延:

$$T_j(v_i) = (|P_j(v_i)| - 1)d_T + \frac{L_j(v_i)}{c} \quad (4)$$

式中, c 是信号在星间链路中的传播速度(下同), d_T 表示数据的一次发送时延。则算法追求的最优即等价于 $T_b(v_e)$ 最小。

将 v_j 到 v_i 的最优路径的时延记作 $D_j(v_i)$, 定义集合 S_b 表示为规划从 v_b 出发的最优路径而考察过的所有节点。初始时, 对 $\forall v_i \in V$, 令 $P_b(v_i) = [v_b, v_i]$ (采用中括号表示链表, 以区别于集合), $D_j(v_i) = 0, S_b = \{v_b\}$ 。算法的具体流程如算法1所示, 其中运算符 \oplus 表示将符号后的元素添加到符号前的链表末尾。

算法1 星间通信路由静态规划算法

Alg. 1 Static programming algorithm for inter-satellite communication routing

```

1. for  $i=0$  to  $n$  do
2.   调用式(4)计算  $T_b(v_i)$ 
3. end for
4. while TRUE do
5.   while  $v_x \in V - S_b$  do
6.     if  $T_b(v_x) = \min_{v_j \in V - S_b} \{T_b(v_j)\}$  then
7.        $S_b \leftarrow S_b \cup \{v_x\}$ 
8.        $D_b(v_x) \leftarrow T_b(v_x)$ 
9.       BREAK
10.    end if
11.  end while
12.  while  $v_m \in V - S_b$  do
13.    调用式(4)计算  $T_x(v_m)$ 
14.    if  $T_b(v_m) > T_b(v_x) + T_x(v_m)$  then
15.       $P_b(v_m) \leftarrow P_b(v_x) \oplus v_m$ 
16.       $T_b(v_m) \leftarrow T_b(v_x) + T_x(v_m)$ 
17.    end if
18.  end while
19.  if  $D_b(v_e) \neq 0$  then
20.    BREAK
21.  end if
22. end while

```

2.2 算法有效性分析

本节将证明算法1的有效性。

证明: 假设算法结束时得到的 $P_b(v_e)$ (记作 $P_\#$, 其时延为 $T_\#$) 并非最优, 真正的从 v_b 到 v_e 的最优路径是 P_* , 其时延为 T_* 。记 v_e 在 P_* 中的前驱节点为 v_i , 显然, P_* 的子结构 $P_b^*(v_i)$ 就是从 v_b 到 v_i 的最优路径(否则 P_* 不是最优的), 将 $P_b^*(v_i)$ 的时延记作 $T_b^*(v_i)$ 。

若算法结束时 $v_i \in S_b$, 在此轮规划进行至算法

1第14行时由 $T_b(v_e) = T_\# > T_* = T_b^*(v_i) + T_1(v_e)$ 可直接得到 P_* , 这与假设矛盾, 所以 $v_i \notin S_b$ 。

若算法结束时 $v_i \in V - S_b$, 记此时 $P_b(v_i)$ 的时延为 T' 。若 $T' \leq T_b^*(v_i)$, 因 $T_b^*(v_i) < T_* < T_\#$, 则 $T' < T_\#$ 。根据算法1第5至7行, 在算法结束前的某轮规划时 v_i 会因满足 $T' = \min_{v_i \in V - S_b} \{T_b(v_i)\}$ 被加入 S_b 中: 这就不符合条件 $v_i \in V - S_b$ 。所以 $T' \leq T_b^*(v_i)$ 不成立。

若 $T' > T_b^*(v_i)$, 说明此时尚未得到 $P_b^*(v_i)$ 。记 v_i 在 $P_b^*(v_i)$ 中的前驱节点为 v_s 。若此时 $v_s \in S_b$, 则在 v_s 加入 S_b 那轮规划进行至算法1第15行时会因 $T_b(v_i) > T_b^*(v_s) + T_s(v_i)$ 而将 v_i 加入 $P_b^*(v_s)$, 将得到 $P_b^*(v_i)$, 与推论矛盾, 故 $v_s \notin S_b$ 。同理, v_s 的前驱节点也不属于 S_b , 否则将推出 $v_s \in S_b$ 。如此推理下去最终可以推知 v_b 在 P_* 上的后继节点 $v_c \notin S_b$ 。但是在满足 $D_b(v_e) = T_\# = \min_{v_i \in V - S_b} \{T_b(v_i)\}$ 时算法结束, 此时仍有 $v_c \notin S_b$, 说明 $T_b(v_e) > T_\#$, 这与假设 $T_\# > T_*$ 矛盾。综上, $T' > T_b^*(v_i)$ 不成立。

前述证明否定了算法结束时有 $v_i \in S_b$ 或 $v_i \in V - S_b$, 于是 $v_i \in \emptyset$, 因此并不存在比 $P_\#$ 更优的 P_* , 从而推翻假设, 证明算法1有效。□

3 求解 ICRPS 的动态规划算法

3.1 基本算法描述

卫星在数据经历时延的同时也在运动着, 卫星位置的改变又导致星间链路的改变, 因此, 若采用算法1求出的从起点 v_b 到终点 v_e 的最优路径 $P_b^\#(v_e)$ 不只包含 v_b 和 v_e , 由于时延的存在, 当数据传送到 $P_b^\#(v_e)$ 中一个异于 v_b 和 v_e 的卫星时, NSSS 的各边权值已经改变, 则 $P_b^\#(v_e)$ 存在失效的可能, 需要重新规划路由。

因此, 本节在算法1的基础上, 设计一种基于动态规划求 ICRPS 可行解的算法2。考虑到智能卫星搭载的宇航级芯片的性能可达 400 MIPS^[16] (今后还会变得更快), 则相比于传送时延, 其路由规划的耗时可以忽略不计, 所以卫星将其规划路由的具体时刻的 NSSS 作为静态网络处理而采用算法1推算其到目的卫星的路由(算法2第6至7行)。可见, 采用算法2求出的路由具有时间属性, 路由的每一段路径都是前驱卫星在其规划路由时刻到后继卫星的最优路径, 这种最优是针对特定时刻的, 不同的规划时刻可能会得到不同的最优路由。

算法 2 求解 ICRPS 的动态规划算法

Alg. 2 Dynamic programming algorithm for ICRPS

1. v_b 建立 NSSS 在当前时刻的邻接矩阵 Q
2. v_b 基于 Q 采用算法 1 求出从 v_b 到 v_e 的最优路由 $P_b^{\#}(v_e)$
3. v_b 将数据发送到 v_b 在 $P_b^{\#}(v_e)$ 中的后继节点 v_s
4. **while** $v_s \neq v_e$ **do**
5. 令原 v_s 成为前驱节点 v_p
6. v_p 更新当前时刻的 Q
7. v_p 基于 Q 采用算法 1 求出从 v_p 到 v_e 的最优路由 $P_p^{\#}(v_e)$
8. v_p 将数据传送到 v_p 在 $P_p^{\#}(v_e)$ 中的后继节点 v_s
9. **end while**

3.2 面向数据分段传送的算法扩展

受网络协议等因素的限制,星间链路一次传输的数据长度往往是有限的,过大的数据将被拆分为若干数据子段以时分复用的方式分别传送,而且两次传送之间还存在一个空时隙,所以卫星在发送每一个数据子段时都必须重新规划当前时刻的路由。

设需要一个长度为 l_T 的数据从 v_b 传送到 v_e ,星间链路一次传输的最大数据长度为 l_m 。考虑有 $l_T/l_m = p, p + 1, \dots, l_r$,则需要将原数据拆分为 $p + 1$ 个数据子段(若 $l_r = 0$ 则拆分为 p 个数据子段),前 p 个数据子段长度为 l_m ,第 $p + 1$ 个数据子段长度为 l_r 。基于此,针对前述分段不连续传送的情况将算法 2 扩展为算法 3。

算法 3 面向数据分段传送的路由算法

Alg. 3 Routing algorithm for segmented data delivery

1. **while** $1 \leq k \leq p + 1$ **do**
2. v_b 建立 NSSS 在当前时刻的邻接矩阵 Q_k
3. v_b 基于 Q_k 采用算法 1 求出从 v_b 到 v_e 的最优路由 $P_b^{[k]}(v_e)$
4. v_b 将第 k 个数据子段发送到 v_b 在 $P_b^{[k]}(v_e)$ 中的后继节点 v_s
5. **while** $v_s \neq v_e$ **do**
6. 令原 v_s 成为前驱节点 v_p
7. v_p 更新当前时刻的 Q_k
8. v_p 基于 Q_k 采用算法 1 求出 v_p 到 v_e 的最优路由 $P_p^{[k]}(v_e)$
9. v_p 将第 k 个数据子段传送到 v_p 在 $P_p^{[k]}(v_e)$ 中的后继节点 v_s
10. **end while**
11. 进入空时隙
12. $k \leftarrow k + 1$
13. **end while**

4 ICRPS 的仿真建模

4.1 智能卫星 Agent 的行为模型

在 ICRPS 的仿真中,一个仿真时刻要同时处理多个智能卫星诸如计算星间链路、收发数据、规划路由等多种行为。为此,采用基于 Agent 的建模与仿真 (Agent-Based Modeling and Simulation, ABMS) 方法^[17],建立智能卫星 Agent (Smart Satellite Agent, SSA) 模型,以刻画智能卫星的自主性与行为并行性。

在 ICRPS 仿真中,存在三种角色的 SSA:源卫星、目的卫星、中转卫星。目的卫星只要接收其他卫星发送的数据即可。

源卫星 v_b 的行为流程如图 1 所示。图 1 中有从 v_b 到目的卫星 v_e 的最优路径 $P_b^{\#}(v_e)$ 不存在的情况,这是因为此时 v_b 与 v_e 无法建链,即 v_b 与 v_e 既没有直接的星间链路也没有经其他卫星中转的链路。为此, v_b 将在规划有效时间 $\Delta T'$ 内反复尝试建链,直至能够建链为止,如果超过 $\Delta T'$ 但 v_b 与 v_e 仍无法建链,则此数据传送任务失败。 $\Delta T'$ 的值取决于任务类型,对于时间敏感任务,尤其是高时效性任务, $\Delta T'$ 仅以秒计。

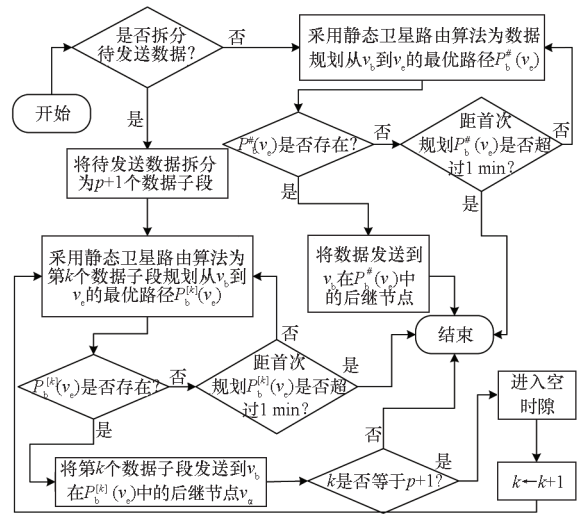


图 1 源卫星的行为流程

Fig. 1 Action process of source satellite

中转卫星 v_a 的行为流程如图 2 所示。中转卫星通过设置中转任务列表 (Transfer Task List, TTL) 来处理多数据子段,每接收到一个数据子段便将其加入 TTL 中。TTL 采用先进先出策略,若 TTL 非空,则调取排在第 1 位的数据子段为其规划路由从而将其发送出去。

在图 2 中,中转卫星也受规划有效时间 $\Delta T'$ 的约束,如果超过 $\Delta T'$ 仍不能建链,则数据传送中断,

需要生成传送中断信息,然后以本星为源卫星、 v_b 为目的卫星,采用算法 3 规划路由将传送中断信息反馈给 v_b ,由 v_b 根据任务时效性决定是否重新传送。如果中转卫星收到其他卫星发来的传送中断信息,将优先处理这些信息。当然,传送中断信息也面临传送失败的风险,必要时可以借助地面站来传送。

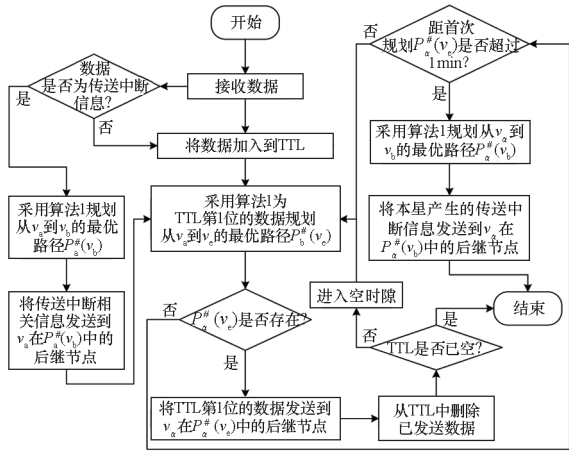


图 2 中转卫星的行为流程

Fig. 2 Action process of relaying satellite

在 Agent 建模时,每一个 SSA 都被赋予上述三种角色的行为,在具体的仿真场景中各 SSA 根据自己被分配的角色而采取相应的行为。

4.2 ICRPS 仿真的调度算法

在仿真执行过程中,每个 SSA 的行为都表示为一个进程,各 SSA 分别拥有一个随时间变化的活动时间序列,可以根据当前 Agent 状态和所处的环境采用逻辑判断方法执行不同的行为。对于多个 SSA 可能试图同时执行行为的情况,则采用进程交互方法进行调度,如算法 4 所示。

算法 4 的关键是为 SSA 设置延迟。延迟是指实体停留在进程中的某一点不再向前移动,延迟结束时实体所在的位置就是复活点——进程继续推进的起点。延迟分为两类:无条件延迟是指预先确定延迟持续的时间,到时间后自动结束延迟;条件延迟是指事先无法确定何时结束延迟,必须等到满足某些条件后进程才能继续。例如空时隙属于无条件延迟,TTL 为空时属于条件延迟。

5 仿真实验与分析

5.1 实验设计

选择体系效能分析仿真 (System-of-systems Effectiveness Analysis Simulation, SEAS) 平台来建立 SSA 模型,采用 ABMS 方法对 ICRPS 进行仿真实验。

算法 4 SSA 仿真调度算法

Alg. 4 Schedule algorithm for SSA simulation

1. 设置仿真结束时间 t_{end}
2. 初始化 SSA 对象,生成 SSA 行为进程
3. 确定 SSA 行为进程的初始复活点及相应的时间值 $T_i (i = 1, 2, \dots, m, \text{反映了进程优先级从高到低})$
4. **while** TRUE **do**
5. 推进仿真时钟 $t_{sim} = \min_{SSA-i \text{ 处于无条件延迟}} \{ T_i \}$
6. **if** $t_{sim} \leq t_{end}$ **then**
7. **for** $i = 1$ **to** m **do**
8. **if** $T_i = t_{sim}$ **then**
9. 从当前复活点开始推进 SSA - i 的进程,直至下一次延迟发生
10. **if** 下一次延迟是无条件延迟 **then**
11. 设置 SSA - i 的进程的下次复活时间 t'_i
12. $T_i \leftarrow t'_i$
13. **end if**
14. **end if**
15. **if** $T_i < t_{sim}$ **then**
16. **if** 满足 SSA - i 进程的延迟结束条件 **then**
17. 从当前复活点开始推进 SSA - i 的进程,直至下一次延迟发生
18. **end if**
19. **if** 下一次延迟是无条件延迟 **then**
20. 设置 SSA - i 的进程的下次复活时间 t'_i
21. $T_i \leftarrow t'_i$
22. BREAK
23. **end if**
24. **end if**
25. **end for**
26. **else**
27. BREAK
28. **end if**
29. **for** $i = 1$ **to** m **do**
30. **if** SSA - i 的状态发生变化 **then**
31. 采用临时状态更新 SSA - i 的状态
32. **end if**
33. **end for**
34. **end while**

针对如表 1 所示的不同 SSS 分别设计 ICRPS 仿真实验。一次实验考察一个以 SSS 中第 i 号卫星 ($0 \leq i \leq 59, i \in N$) 为源卫星、第 j 号卫星 ($0 \leq j \leq 59, j \in N, i \neq j$) 为目的卫星的卫星对之间的路由,则对每种 SSS 进行 $A_{60}^2 = 3\ 540$ 次实验,实验数量

共计 10 620 次。

表 1 SSS 类型
Tab. 1 Types of SSS

SSS 种类	星座构型	卫星编号
SSS - 1	Walker - δ 30/10/5 : 52°, 16 801.1 km	0 - 29
	Walker - δ 30/10/5 : 52°, 661.955 km	30 - 59
SSS - 2	Walker - δ 60/10/5 : 52°, 661.955 km	0 - 59
SSS - 3	Walker - δ 60/10/5 : 52°, 16 801.1 km	0 - 59

取 $\gamma_{i,\min} = \gamma_{j,\min} = 20^\circ, \gamma_{i,\max} = \gamma_{j,\max} = 60^\circ$ ^[18], $h_1 = 120 \text{ km}$ ^[19], $R_E = 6\,378.137 \text{ km}$ 。数据长度为 2 048 000 bit, 设星间链路一次传输的最大数据长度为 51 200 bit, 即源卫星需要分 40 次发送长度为 51 200 bit 的数据子段。卫星发送数据的速率为 100 kbit/s, 空时隙为 0.1 s, 规划有效时间 $\Delta T' = 1 \text{ min}$ 。仿真开始时刻为格林尼治时间 2019 年 9 月 19 日 0 时 0 分 2 秒, 持续时间为 2 min, 步长为 1 ms。

5.2 实验结果分析

5.2.1 实验结果分类

将 ICRPS 仿真实验可能出现的结果划分为图 3 所示分类树中实线框标识的 7 种类型, 据此对 10 620 个实验结果进行分类, 如表 2 所示。

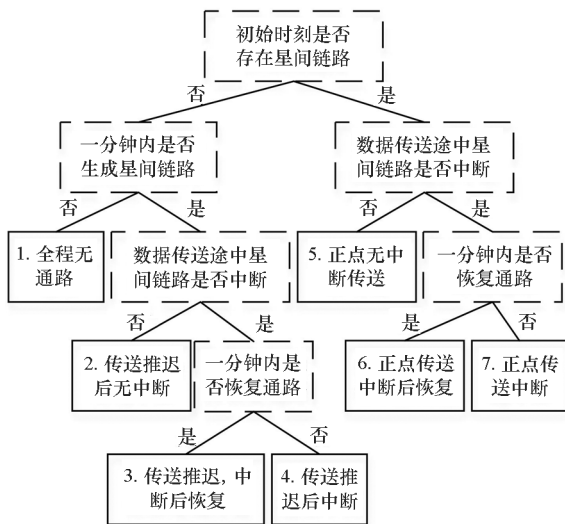


图 3 实验结果的分类树

Fig. 3 Classification tree of experiment results

可见, 在相同的星间链路参数下不同 SSS 的实验结果并不相同。中轨卫星组成的 SSS - 3 任意两个成员之间的数据传送都能够正点连续完成, 没有任何推迟或中断, 因为中轨卫星高度较大, 姿态更加灵活, 更易满足建立星间链路的

条件。

表 2 实验结果分类统计

Tab. 2 Statistics about classification of experiment results

SSS 种类	实验结果分类次数						
	1 型	2 型	3 型	4 型	5 型	6 型	7 型
SSS - 1	2 601	55	0	14	870	0	0
SSS - 2	2 778	245	20	25	385	36	51
SSS - 3	0	0	0	0	3 540	0	0

SSS - 1 和 SSS - 2 的实验结果绝大多数都全程无通路, 因为其中包含的低轨卫星高度较低, 很多情况下与其他低轨卫星或中轨卫星并不可见或无法通信, 遂不能直接建链。事实上, 在 SSS - 2 中, 甚至有 20 颗卫星 (编号为 6、8、9、11、19、20、22、23、30、31、33、34、42、44、45、47、55、56、58、59) 与其他所有成员在 2 min 内都不能建链。

SSS - 1 的结果更能反映卫星轨道对星间路由的影响: 870 次正点连续传送都发生在中轨卫星之间; 55 次推迟连续发送和 14 次传送推迟后中断都在低轨卫星之间; 2 601 次全程无通路中没有一次发生在中轨卫星之间。

不过, 图 3 中的“无法建链”仅表示从仿真开始后 1 min 以内无法建链, 并不意味着卫星对之间永远不能建链, 例如 SSS - 2 的 0 号成员与 3 号成员在仿真开始后 3.094 min 才成功建链, 超过了 1 min 的规划有效时间。

上述结果都是基于 5.1 节中的星间链路参数取值, 并不是所有轨道类型的卫星在这些参数约束下都表现良好, $h_1, \gamma_{i,\min}, \gamma_{j,\min}, \gamma_{i,\max}, \gamma_{j,\max}, \Delta T'$ 等参数对低轨卫星的建链及规划结果的影响远大于对中轨卫星的影响。如何设置相关参数以避免出现数据传送失败及中断是一个重要的问题, 仅在路由算法层面是不足以解决的, 需要对卫星本身进行改进, 例如增大卫星天线扫描范围^[20]、卫星姿态机动、改变卫星星座构型甚至卫星轨道。这种改进需要符合实际, 不能为了追求高效能而随意放宽约束。

5.2.2 路由跳数分布

10 620 个实验中有 5 109 个卫星对完成全部 40 个数据子段的传送, 因每个数据子段都对应一个路由, 共得到 204 360 个有效路由, 这些路由跳数的分布如图 4 所示。

SSS - 1 和 SSS - 3 的有效路由跳数分布在结构上相似, 都是跳数为 1 和 2 的路由占绝大多数, 在 5.2.3 节将介绍何时出现跳数为 3 的路由, 而

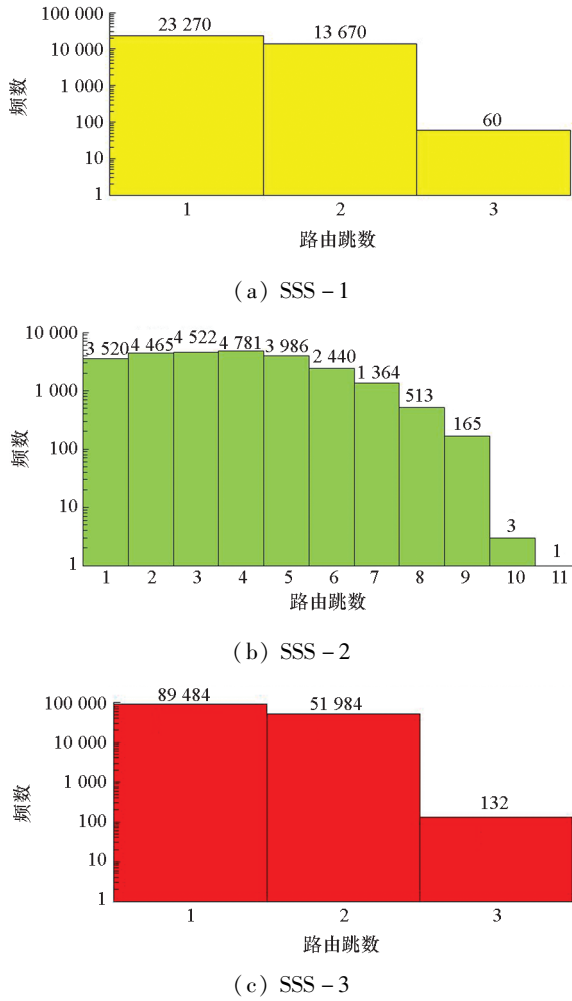


图 4 有效路由跳数分布直方图

Fig. 4 Histograms of normal routing hops

出现跳数大于 1 的路由则说明即使中轨卫星之间也会有无法直接通信的情况。

SSS-2 的有效路由则出现了 1 到 11 这几种跳数,全部 25 760 个有效路由的跳数均值为 3.683 54,跳数为 1 的路由只占 14%。如前所述,低轨卫星较难直接建链,所以 SSS-2 成员之间直接通信的占比明显小于 SSS-1 和 SSS-3,多数卫星对的星间通信还需通过中转卫星来完成。

5.2.3 路由的改变

按卫星对间传送不同数据子段的路由是否有变化可以对图 3 中的第 2、3、5、6 种结果类型分别进一步细分,如图 5 所示。

基于图 5 对表 2 中的部分实验结果进行二次分类,其结果如表 3 所示。表 3 中 3-2 型和 6-2 型的结果都不为零,说明采用算法 3 能够在星间通信中断的情况下重新恢复路由;而表 3 中 3-1 列和 6-1 列的结果都为零则表明不存在传送中断后恢复但路由不变的情况,因为中断前后的

NSSS 并不相同。

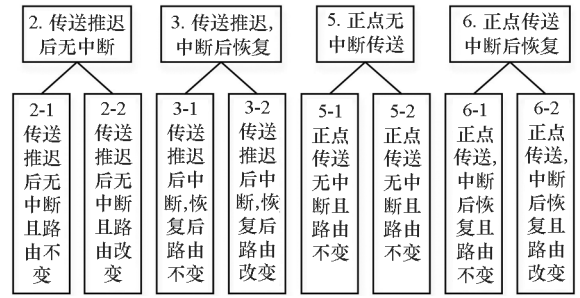


图 5 实验结果的二次分类

Fig. 5 Further classifications of experiment results

表 3 实验结果的二次分类统计

Tab. 3 Statistics about further classifications of experiment results

SSS 种类	实验结果分类次数							
	2-1 型	2-2 型	3-1 型	3-2 型	5-1 型	5-2 型	6-1 型	6-2 型
SSS-1	55	0	0	0	840	30	0	0
SSS-2	110	135	0	20	251	134	0	36
SSS-3	0	0	0	0	3 474	66	0	0

表 3 中路由改变的实验结果共计 421 个,它们的路由变化类型可以大致分为 7 种,如图 6 所示。类型 1、2、3(图 6 中三者部分点线完全重合)见于 SSS-1 和 SSS-3 的中轨卫星对路由的变化。图 4(a)和图 4(c)中所有跳数为 3 的数据子段路由都出现在类型 1 所代表的卫星对路由中。类型 3 表示路由改变但跳数不变,仅见于跳数为 2 的路由。

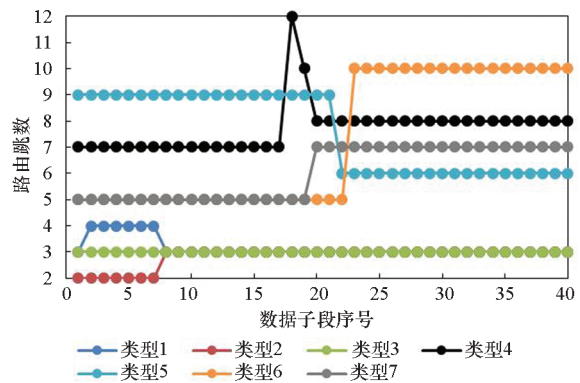


图 6 路由变化类型

Fig. 6 Types of routing changes

类型 4、5、6、7(图 6 中类型 6 和类型 7 的部分点线完全重合)见于 SSS-2 的低轨卫星对的路由变化,其跳数有的由大变小(类型 5),有的由小变大(类型 6、7),还有先增后减(类型 4)。类

型 7 是无中断传送的路由,其跳数变化幅度相对较小,而类型 4、5、6 都是中断后恢复的路由,其跳数变化幅度较大。

图 7 和图 8 分别展示了上述 7 种变化路由及不变路由(类型 8)的狭义时延和广义时延。狭义时延是指发送时延与传输时延之和。广义时延是指狭义时延加上其他形式的时延,主要有两种:排队时延是指数据为等待排在其之前的数据都发送完而耗费的时间;中断时延是指数据等待中断的星间链路恢复而耗费的时间。如图 7 所示,狭义时延与跳数的变化趋势几乎一致,表明二者存在相关性。

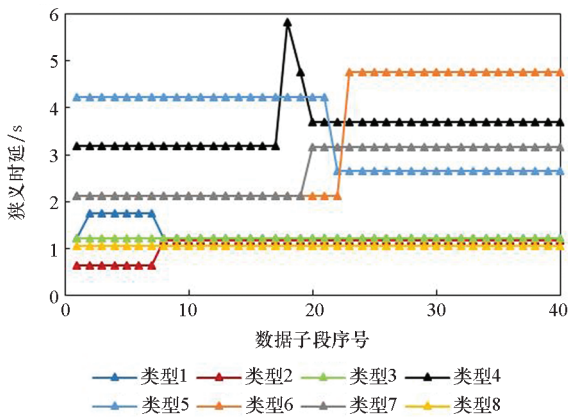


图 7 典型路由的狭义时延

Fig. 7 Simple delays of typical routings

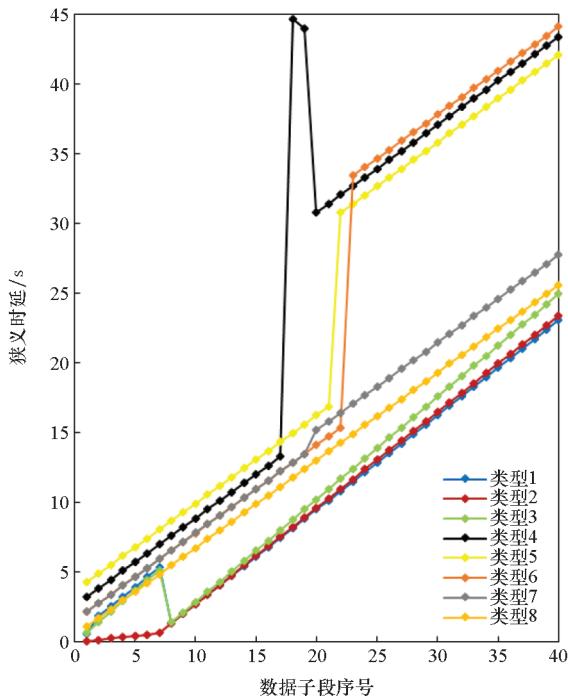


图 8 典型路由的广义时延

Fig. 8 General delays of typical routings

广义时延的变化如图 8 所示,呈现以下规律:

1) 数据子段 1 的广义时延总等于狭义时延,

此后随着数据子段序号增大,其广义时延一般会随之递增,因为越靠后的数据子段经历的排队时延越大,如类型 2、8;

2) 规律 1 建立在所有数据子段路由的前两个卫星(即源卫星及其后继卫星)都相同的前提下,若源卫星的后继卫星改变,则排队时延将从头算起,就会出现类型 1、3 所示的广义时延起伏变化的情况(图 8 中类型 1 的部分点线被类型 2、3 遮挡);

3) 相比于无中断传送时跳数增加对广义时延的影响(类型 7),中断时延对广义时延的影响程度更大,类型 4、5、6 的路由中断时延甚至超过了一些数据子段的广义时延。

路由变化的原因可以分为两类:

1) 数据连续传送中存在时延更短的路由:包括类型 1 中跳数由 3 变为 2、类型 3 中跳数由 9 变为 7,及类型 2 的路由变化;

2) 原星间链路失效,重新规划路由:包括类型 2、5、6、7,及类型 1 中跳数由 2 变为 3、类型 3 中跳数由 6 变为 11。

不过,类型 5 所示的路由变化表明,原因 2 导致的路由变化并不必然使狭义时延增大,时延变短的路由变化也不一定是原因 1 造成的。

6 结论

本文在提出求解 ICRPS 的动态规划算法的基础上,结合智能卫星所特有的自主性,将智能卫星建模为 Agent,并为基于智能卫星 Agent 的 ICRPS 仿真设计了调度算法。借助 SEAS 平台对三种 SSS 内部成员之间的路由进行了仿真实验,实验结果表明,本文研究的算法可以有效解决不同 SSS 下的路由问题。不过,对实验结果分析发现,路由规划的结果受到 SSS 体系架构和硬件层面因素的深刻影响,如 SSS 星座构型和卫星轨道,以及星间链路的数据传输能力等,如何优化这些因素以提升 SSS 的效能将是今后的重要研究方向。

参考文献 (References)

[1] GURFIL P, HERSCOVITZ J, PARIENTE M. The SAMSON project—cluster flight and geolocation with three autonomous nano-satellites[C]//Proceedings of the 26th Annual AIAA/USU Conference on Small Satellites, 2012.

[2] TRUSZKOWSKI W, HALLOCK H, ROUFF C, et al. Autonomous and autonomic systems: with applications to NASA intelligent spacecraft operations and exploration systems[M]. London: Springer London, 2010.

[3] XU W C, JIANG M, TANG F L, et al. Network coding-

- based multi-path routing algorithm in two-layered satellite networks[J]. IET Communications, 2018, 12(1): 2-8.
- [4] XIE P, ZHANG Z S, ZHANG J. Inter-satellite routing algorithm by searching the global neighborhood for dynamic inter-satellite networks [C]//Proceedings of the Tenth International Conference on Advanced Computational Intelligence, 2018: 673-678.
- [5] HUANG J H, LIU W X, SU Y X, et al. Cascade optimization design of inter-satellite link enhanced with adaptability in future GNSS satellite networks [J]. GPS Solutions, 2018, 22(2): 44.
- [6] WANG F, JIANG D D, QI S. An adaptive routing algorithm for integrated information networks[J]. China Communications, 2019, 16(7): 195-206.
- [7] 尤启迪, 王小云, 金星虎, 等. 面向信息快速回传的卫星网络多路径路由算法[J]. 武汉大学学报(工学版), 2019(8): 747-752.
- YOU Q D, WANG X Y, JIN X H, et al. Multipath routing algorithm of information fast returned satellite network[J]. Engineering Journal of Wuhan University, 2019(8): 747-752. (in Chinese)
- [8] SORET B, SMITH D. Autonomous routing for LEO satellite constellations with minimum use of inter-plane links [C]//Proceedings of IEEE International Conference on Communications, 2019: 1-6.
- [9] ZHANG L, YAN F, ZHANG Y Y, et al. A routing algorithm based on link state information for LEO satellite networks[C]//Proceedings of IEEE GLOBECOM Workshops, 2020: 1-6.
- [10] ROTH M, BRANDT H, BISCHL H. Implementation of a geographical routing scheme for low earth orbiting satellite constellations using intersatellite links [J]. International Journal of Satellite Communications and Networking, 2021, 39(1): 92-107.
- [11] 梁俊, 孙伟超, 肖楠, 等. 基于切比雪夫神经网络的软件定义卫星网络智能路由策略[J]. 国防科技大学学报, 2020, 42(5): 23-30.
- LIANG J, SUN W C, XIAO N, et al. Intelligent routing strategy for software-defined satellite network based on Chebyshev neural network[J]. Journal of National University of Defense Technology, 2020, 42(5): 23-30. (in Chinese)
- [12] 谢希仁. 计算机网络[M]. 北京: 电子工业出版社, 2017.
- XIE X R. Computer network[M]. Beijing: Publishing House of Electronics Industry, 2017. (in Chinese)
- [13] 汪小帆, 李翔, 陈关荣. 网络科学导论[M]. 北京: 高等教育出版社, 2012.
- WANG X F, LI X, CHEN G R. Network science: an introduction [M]. Beijing: Higher Education Press, 2012. (in Chinese)
- [14] 张洪波. 航天器轨道力学理论与方法[M]. 北京: 国防工业出版社, 2015.
- ZHANG H B. Theories and methods of spacecraft orbital mechanics [M]. Beijing: National Defense Industry Press, 2015. (in Chinese)
- [15] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1(1): 269-271.
- [16] 徐帅, 林宝军, 刘迎春, 等. 基于龙芯宇航级芯片的BSP开发和移植[J]. 计算机工程与科学, 2020, 42(4): 571-579.
- XU S, LIN B J, LIU Y C, et al. Development and transplantation of BSP based on Godson aerospace chip[J]. Computer Engineering & Science, 2020, 42(4): 571-579. (in Chinese)
- [17] MACAL C M, NORTH M J. Tutorial on agent-based modelling and simulation [J]. Journal of Simulation, 2010, 4(3): 151-162.
- [18] 王东会, 刘文祥, 唐小妹, 等. 一种考虑异常中断的导航星座星间链路路由改进算法[J]. 中南大学学报(自然科学版), 2014, 45(3): 762-768.
- WANG D H, LIU W X, TANG X M, et al. Improvement of navigation inter-satellite link routing algorithm considering abnormal link disruption [J]. Journal of Central South University (Science and Technology), 2014, 45(3): 762-768. (in Chinese)
- [19] 闫野, 周伯昭, 任萱. 关于卫星网整网自主定轨方法的探讨[J]. 宇航学报, 2002, 23(2): 80-83.
- YAN Y, ZHOU B Z, REN X. A discussion on satellite-network autonomous orbit determination [J]. Journal of Astronautics, 2002, 23(2): 80-83. (in Chinese)
- [20] 黄今辉. 卫星导航系统星间链路网络优化设计技术研究[D]. 长沙: 国防科技大学, 2018.
- HUANG J H. Research on the design and optimization of inter-satellite link networks in global navigation satellite systems [D]. Changsha: National University of Defense Technology, 2018. (in Chinese)