KATHOLIEKE UNIVERSITEIT
LEUVEN

Faculty of Business and Economics

A note on peer-to-peer satellite refueling strategies

Sofie Coene, Atri Dutta, Frits Spieksma and Panagiotis Tsiotras

DEPARTMENT OF DECISION SCIENCES AND INFORMATION MANAGEMENT (KBI)

KBI 1003

A note on peer-to-peer satellite refueling strategies*

Sofie Coene† Atri Dutta‡
Frits C.R. Spieksma† Panagiotis Tsiotras‡

Abstract

We revisit the peer-to-peer refueling problem, in which the maneuvering satellites are allowed to interchange their orbital positions [2]. We show that the problem is computationally hard, by reducing it to a special case of the three-index assignment problem. On the positive side, we show that the size of instances from practice is such that a state-of-the-art integer programming solver is able to find optimal solutions in little computing time.

Keywords: three-index assignment problem; complexity; integer programming.

1 Introduction

The following problem was described in Dutta and Tsiotras [2]. Given are n satellites distributed over n slots in a circular orbit. For the sake of simplicity, we consider that satellite s_i occupies the slot ϕ_i, for all $i \in \{1, 2, \ldots, n\}$. Each satellite is denoted as either *fuel-deficient*, or *fuel-sufficient*; this depends on whether the satellite has at least a minimum required amount of fuel. The idea put forward in [2] is that the satellites can redistribute the fuel among themselves by engaging in fuel exchange in pairs; for each pair, a satellite moves (performs an orbital transfer) to rendezvous with another satellite, exchanges fuel, and returns back to any available slot. This redistribution of available fuel among the satellites is an integral part of a mixed refueling strategy, which is much more fuel-efficient than supplying fuel via a single service-vehicle, particularly with an increasing number of satellites in the constellation [2].

We say that a pair of satellites is a *feasible* pair when (i) the pair consists of a fuel-sufficient satellite and a fuel-deficient satellite, and (ii) the amount of

*This research was supported by OT Grant OT/07/015.

†Operations Research Group, Katholieke Universiteit Leuven, Naamsestraat 69, B-3000 Leuven, Belgium. Email: {sofie.coene;frits.spieksma}@econ.kuleuven.be.

‡Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, USA. Email: { a3d@;tsiotras@ }gatech.edu.

fuel carried by the two satellites of the pair can be redistributed among them in a way that each satellite of the pair has at least the required amount of fuel at the end of the refueling process.

The task of redistributing fuel is now accomplished by considering movements of the satellites that we represent by an ordered triple (i, j, k): this means that satellite s_i at position ϕ_i moves to rendezvous with satellite s_j at position ϕ_j, and after undergoing a fuel exchange, satellite s_i moves to position ϕ_k (while satellite s_j stays at position ϕ_j). Clearly, we only take into account those triples that contain a feasible pair of satellites. Observe that the moving satellite can be fuel-sufficient (in which case this satellite will donate fuel to its companion) or fuel-deficient (in which case the satellite will receive fuel from its companion). There is a given cost-coefficient $c(i, j, k)$ associated with each triple. These cost-coefficients correspond to the amount of fuel expended during the orbital transfers; notice that this depends on a number of aspects, such as the angle of separation of the satellites, the mass and engine of the maneuvering satellite, etc. Notice that it might happen that the amount of fuel of the two satellites making up the feasible pair is not sufficient for the moving satellite to reach some position ϕ_k. In that case, the coefficient $c(i, j, k)$ simply receives a large value; we come back to this issue of fuel cost extensively in Section 2. The goal is to find triples (i, j, k) such that

(i) each satellite is in at most one triple,

(ii) each satellite that is fuel-deficient is in at least one triple with a fuel-sufficient satellite,

(iii) at the end of refueling process, each position is occupied by at most one satellite, and

(iv) total fuel costs are minimal.

We will refer to this problem as the peer-to-peer refueling problem, and a set of triples that satisfies (i), (ii), and (iii) is said to be a feasible solution to the problem. Clearly, for a solution to exist, the number of fuel-deficient satellites should not exceed the number of fuel-sufficient satellites.

In this note

- we establish the computational complexity of the peer-to-peer refueling problem, and
- we show that the size of the instances that occur in practice allow using a state-of-the-art integer programming solver to solve these instances to optimality.

2 Problem Statement

As described in Section 1, our problem is to decide which satellites pair up for fuel exchanges, and to which positions the active satellites return, in order

to minimize fuel costs. In this problem description, we allow explicitly for a satellite to end its movement in a position that is different from its starting position. Clearly, one could restrict the solutions to be such that starting and ending slots need to be identical, that is, $\phi_i = \phi_k$. Although this would make the problem easy (see Section 3), it has been illustrated that this might lead to suboptimal solutions, i.e., solutions with a larger than necessary fuel cost [2].

Let us consider a triple (i, j, k) and the associated cost $c(i, j, k)$. The movement of the active satellite s_i consists of two trips: the first trip, when it moves from its position ϕ_i to another position ϕ_j harboring satellite s_j, and the second trip, where satellite s_i travels from position ϕ_j to position ϕ_k. Let p^i_{ij} denote the fuel expended by the satellite s_i when it transfers from slot ϕ_i to slot ϕ_j. The fuel consumed by satellite i in the first part can then be expressed as (see [2]):

$$p^i_{ij} = (m_{\text{si}} + f_i^-)(1 - e^{-\Delta V_{ij}/g_0 I_{\text{spi}}}), \tag{1}$$

where

- m_{si} refers to the mass of the permanent structure of satellite s_i (fuel not included), $1 \leq i \leq n$,
- f_i^- refers to the initial amount of fuel present in satellite i, $1 \leq i \leq n$,
- I_{spi} refers to the specific impulse of engine of satellite s_i, $1 \leq i \leq n$,
- ΔV_{ij} is the velocity change required to transfer from slot ϕ_i to slot ϕ_j, for each ordered pair of positions ϕ_i and ϕ_j, $1 \leq i, j \leq n$, and
- g_0 is a constant denoting the acceleration due to gravity at the Earth's surface.

In order to express the fuel consumption of an active satellite s_i in the second part of its maneuver, we need to know how much fuel is present in satellite s_i after the exchange. This quantity, however, is chosen such that the amount of fuel present in the active satellite directly after the exchange is the minimum amount needed so that satellite s_i can return from position ϕ_j to position ϕ_k. (This follows from the fact that there is no point in moving fuel (mass) which is not needed). Now, by letting p^i_{jk} stand for the fuel consumption of satellite s_i when traveling from position ϕ_j to position ϕ_k, we can write down the following equation:

$$p^i_{jk} = (m_{\text{si}} + \underline{f}_i + p^i_{jk})(1 - e^{-\Delta V_{jk}/g_0 I_{\text{spi}}}),$$

where $\underline{f}_i$ refers to the amount of fuel required for satellite s_i after it has reached its position following the exchange.

Solving this equation for p^i_{jk} gives:

$$p^i_{jk} = (m_{\text{si}} + \underline{f}_i)(e^{\Delta V_{jk}/g_0 I_{\text{spi}}} - 1). \tag{2}$$

Summing equations (1) and (2), we derive an explicit expression for our cost-coefficients $c(i, j, k)$:

$$c(i,j,k) = p^i_{ij} + p^i_{jk} = \\ m_{\text{si}}(e^{\Delta V_{jk}/g_0 I_{\text{spi}}} - e^{-\Delta V_{ij}/g_0 I_{\text{spi}}}) + f_i^-(1 - e^{-\Delta V_{ij}/g_0 I_{\text{spi}}}) + \underline{f}_i(e^{\Delta V_{jk}/g_0 I_{\text{spi}}} - 1). \tag{3}$$

The problem of peer-to-peer refueling seeks to find the set of feasible triples with minimum total fuel costs. In order to analyze the computational complexity of the problem, we next describe precisely what constitutes the input to the peer-to-peer refueling decision problem. For each satellite $i \in \{1, 2, \ldots, n\}$, we have:

- m_{si}: the mass of satellite i,
- f_i^-: the amount of fuel present in satellite i,
- I_{spi}: the thrust of satellite i,
- $\underline{f}_i$: the amount of fuel needed for satellite i to reach its estimated lifespan (not including fuel needed for movements),

and for each pair of positions $i, j \in \{1, 2, \ldots, n\}$, we have:

- ΔV_{ij}: the velocity change required to go from position i to position j,

and finally, we have:

- L: an integer, denoting an upper bound on the total fuel cost of a solution.

Given this input, the decision version of the peer-to-peer refueling problem seeks to find the answer to the following question: does there exist a feasible set of triples such that total fuel costs (as expressed by (3)) do not exceed L?

3 The complexity of peer-to-peer refueling

Computational complexity is a field that attempts to establish the hardness of optimization problems, we refer to Garey and Johnson [3] for a classical introduction.

In this section we prove that peer-to-peer refueling is NP-hard, by exhibiting a reduction from a special 3-dimensional assignment problem, which was introduced in, and proven to be NP-hard by Burkard et al. [1]. The implication of this result is that it is unlikely (meaning that it would imply P=NP) that there exists an efficient algorithm solving each instance of peer-to-peer refueling exactly.

> Problem: 3-dimensional assignment with decomposable coefficients (3AP-DC)
>
> Input: $3q$ nonnegative numbers a_i, b_i and c_i, for $i = 1, 2, \ldots, q$ and

an integer K.

Question: do there exist two permutations $\pi : \{1, 2, \ldots, n\} \mapsto \{1, 2, \ldots, n\}$ and $\sigma : \{1, 2, \ldots, n\} \mapsto \{1, 2, \ldots, n\}$ such that $\sum_{i=1}^{q} a_i b_{\pi(i)} c_{\sigma(i)} \leq K$?

We now show that peer-to-peer refueling is at least as hard as 3AP-DC. We build an instance of our peer-to-peer refueling problem that consists of q fuel-sufficient satellites (indexed by $i = 1, 2, \ldots, q$), as well as q fuel-deficient satellites (indexed by $i = q+1, q+2, \ldots, 2q$). Hence $n := 2q$. We now specify all input parameters (see Section 2), using the symbol M for some large value. For each satellite $i = 1, \ldots, n$, we set:

$$f_i^- := M, I_{\text{spi}} := 1.$$

For each fuel-sufficient satellite $i = 1, 2, \ldots, q$, we set:

$$\underline{f}_i := 0, m_{\text{si}} := a_i.$$

For each fuel-deficient satellite $i = q+1, q+2, \ldots, 2q$, we set:

$$\underline{f}_i := M + 1, m_{\text{si}} := M.$$

For each pair of positions (i, j) where position i harbors a fuel-sufficient satellite ($1 \leq i \leq q$), and position j harbors a fuel-deficient satellite ($q + 1 \leq j \leq 2q$), we set

$$\Delta V_{ij} := 0, \text{ and } \Delta V_{ji} := g_0 \cdot \ln(b_j c_i + 1).$$

Finally, we set $L := K$.

This completes the description of the instance of the peer-to-peer refueling problem. Notice that we have allowed for non-symmetric ΔV_{ij} values.

Let us now argue that there exists a 1-to-1 correspondence between yes-instances of 3AP-DC and the peer-to-peer refueling problem. Suppose that the instance of 3AP-DC admits a yes-answer, i.e., there exist permutations π and σ such that $\sum_{i=1}^{q} a_i b_{\pi(i)} c_{\sigma(i)} \leq K$. Then we simply copy that solution to the peer-to-peer refueling problem: fuel-sufficient satellite i ($1 \leq i \leq q$) moves to fuel-deficient satellite $q + \pi(i)$, and next returns to position $\sigma(i)$. The costs of these movements follow from substituting the values given above in (3), leading to

$$\begin{aligned} &c(i, j, k) = \\ &= m_{\text{si}}(e^{\Delta V_{jk}/g_0 I_{\text{spi}}} - e^{-\Delta V_{ij}/g_0 I_{\text{spi}}}) + f_i^-(1 - e^{-\Delta V_{ij}/g_0 I_{\text{spi}}}) + \underline{f}_i(e^{\Delta V_{jk}/g_0 I_{\text{spi}}} - 1) \\ &= a_i(e^{\ln(b_{\pi(i)+q} c_{\sigma(i)}+1)} - 1) = a_i b_{\pi(i)} c_{\sigma(i)}. \end{aligned}$$

Hence, if there exist permutations π and σ such that $\sum_{i=1}^{q} a_i b_{\pi(i)} c_{\sigma(i)} \leq K$, then there is a solution with cost bounded by L for the peer-to-peer refueling problem.

Suppose now that the instance of peer-to-peer refueling has a solution with a cost no more than L. Notice that the fuel-deficient satellites cannot move due to their large mass $m_{\rm si} = M$; moving any fuel deficient satellite would lead to costs exceeding L. Thus, we can define the permutation π, by inspecting which fuel sufficient satellite i visits which fuel deficient satellite $\pi(i) := j - q$, and we can find permutation σ by verifying to which position $\sigma(i) := k$ each satellite i returns. The resulting solution to 3AP-DC will have cost no more than K. All this leads to:

Theorem 1 *The peer-to-peer refueling problem is NP-hard.*

Clearly, the difficulty of peer-to-peer refueling changes when we restrict the solutions such that each satellite i must return to its original position i, $1 \leq i \leq n$. The costs then simplify to:

$$c(i,j,i) = p^i_{ij} + q^i_{jk} =$$
$$= m_{\rm si}(e^{\Delta V_{ji}/g_0 I_{\rm spi}} - e^{-\Delta V_{ij}/g_0 I_{\rm spi}}) + f_i^-(1 - e^{-\Delta V_{ij}/g_0 I_{\rm spi}}) + \underline{f}_i(e^{\Delta V_{ji}/g_0 I_{\rm spi}} - 1),$$

and no longer depend on k. Hence, by modeling the resulting problem as an assignment problem with costs $d(i,j) = \min(c(i,j,i), c(j,i,j))$ between the fuel sufficient satellites i and the fuel deficient satellites j, it is clear that efficient algorithms exist for this special case of the problem (see e.g. Schrijver [4]).

4 Computational results using an Integer Programming formulation

Let us now write down an integer programming model for the peer-to-peer refueling problem. To simplify notation, we use FD (FS) to denote the set of fuel-deficient (fuel-sufficient) satellites; we let $S \equiv FD \cup FS$ stand for the set of all satellites. We use a decision variable

$$x_{ijk} = \begin{cases} 1 & \text{if satellite } i \text{ moves to satellite } j \text{ and ends up in position } k; \\ 0 & \text{otherwise.} \end{cases}$$

Recall that we only define variables for those triples that contain a feasible pair, and whose corresponding maneuver is called *feasible* in [2]; for instance, variables of the form x_{ijj} do not exist.

$$\begin{array}{lcll} \min & \sum_{i,j\in S}\sum_{k=1}^{n} c(i,j,k)x_{ijk} & & (4) \\ \text{s.t.} & \sum_{j\in FS}\sum_k x_{jik} + \sum_{j\in FS}\sum_k x_{ijk} = 1 & \text{for all } i \in FD & (5) \\ & \sum_{j\in FD}\sum_k x_{ijk} + \sum_{j\in FD}\sum_k x_{jik} \leq 1 & \text{for all } i \in FS & (6) \\ & \sum_{i\in S}\sum_{j\in S} x_{ijk} - \sum_{j\in S}\sum_\ell x_{kj\ell} \leq 0 & \text{for all } k & (7) \\ & x_{ijk} \in \{0,1\} & \text{for all } i,j,k. & (8) \end{array}$$

Notice that:

- the cost-coefficients in the objective function (4) are defined by (3),
- constraints (5) express that each fuel-deficient satellite is in one triple with a fuel-sufficient satellite,
- constraints (6) express that each fuel-sufficient satellite is in at most one triple,
- constraints (7) express that each position can harbor at most one satellite; indeed, if some satellite ends up in position k, then the satellite starting at that position must have moved,
- notice that in case the number of fuel-deficient satellites equals the number of fuel-sufficient satellites, the resulting constraints become exactly those from an axial three-index assignment problem (with the cost-coefficients displaying a specific structure, see Spieksma [5] for an overview of other possible structures).

Label	Description
C_1	Altitude $= 35,786$ Km, $n = 10$, $T = 12$, $f_i = 30$, $\underline{f}_i = 12$, $m_{si} = 70$ f_i^-: $30, 30, 6, 6, 6, 6, 6, 30, 30, 30$
C_2	Altitude $= 1,200$ Km, $n = 16$, $T = 30$, $f_i = 30$, $\underline{f}_i = 15$, $m_{si} = 70$ f_i^-: $30, 30, 30, 30, 30, 30, 10, 10, 10, 10, 10, 10, 10, 10, 30, 30$
C_3	Altitude $= 2,000$ Km, $n = 12$, $T = 30$, $f_i = 30$, $\underline{f}_i = 15$, $m_{si} = 70$ f_i^-: $30, 30, 30, 10, 10, 10, 10, 10, 10, 30, 30, 30$
C_4	Altitude $= 6,000$ Km, $n = 18$, $T = 25$, $f_i = 25$, $\underline{f}_i = 12$, $m_{si} = 75$ f_i^-: $25, 25, 25, 25, 25, 25, 25, 25, 25, 6, 6, 6, 6, 6, 6, 6, 6, 6$
C_5	Altitude $= 12,000$ Km, $n = 12$, $T = 20$, $f_i = 25$, $\underline{f}_i = 12$, $m_{si} = 75$ f_i^-: $25, 25, 25, 25, 25, 25, 8, 8, 8, 8, 8, 8$
C_6	Altitude $= 1,400$ Km, $n = 14$, $T = 35$, $f_i = 25$, $\underline{f}_i = 12$, $m_{si} = 75$ f_i^-: $25, 25, 25, 25, 25, 25, 25, 8, 8, 8, 8, 8, 8, 8$
C_7	Altitude $= 30,000$ Km, $n = 16$, $T = 15$, $f_i = 30$, $\underline{f}_i = 15$, $m_{si} = 70$ f_i^-: $10, 10, 10, 10, 10, 10, 10, 10, 28, 28, 28, 28, 28, 28, 28, 28$
C_8	Altitude $= 1,200$ Km, $n = 16$, $T = 30$, $f_i = 30$, $\underline{f}_i = 15$, $m_{si} = 70$ f_i^-: $30, 10, 30, 10, 30, 10, 30, 10, 30, 10, 30, 10, 30, 10, 30, 10$
C_9	Altitude $= 2,000$ Km, $n = 36$, $T = 30$, $f_i = 30$, $\underline{f}_i = 15$, $m_{si} = 70$ f_i^-: $8, 30, 8, 30, 8, 30, 8, 30, 8, 30, 8, 30, 8, 30, 8, 30, 8, 30,$ $8, 30, 8, 30, 8, 30, 8, 30, 8, 30, 8, 30, 8, 30, 8, 30, 8, 30$
C_{10}	Altitude $= 2,000$ Km, $n = 40$, $T = 30$, $f_i = 25$, $\underline{f}_i = 12$, $m_{si} = 75$ f_i^-: $25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25,$ $6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6$

Table 1: Sample Constellations

The model (4)-(8) was implemented in ILOG OPL Development Studio 5.2, and run on a personal computer with a 2.20 GHz Intel Core 2 Duo processor

and 2.00 GB of RAM. We experimented with 10 instances whose characteristics can be found in Table 1; notice that in each of these instances, the number of fuel-sufficient satellites equals the number of fuel-deficient satellites.

Constellation	n	Opt	Running time	E-P2P	restricted
C1	10	18.515 (i)	0.078	19.11 (3.2%)	26.072 (41%)
C2	16	24.356 (i)	0.172	24.82 (1.9%)	37.478 (54%)
C3	12	18.452 (i)	0.156	18.87 (2.3%)	26.428 (43%)
C4	18	25.787 (i)	0.266	26.26 (1.8%)	40.728 (58%)
C5	12	18.792 (i)	0.125	18.86 (0.36%)	28.385 (51%)
C6	14	19.025 (i)	0.156	19.26 (1.2%)	28.774 (51%)
C7	16	22.539 (i)	0.234	22.75 (0.94%)	34.976 (55%)
C8	16	9.0826 (i)	0.203	10.18 (12%)	9.3751 (3,2%)
C9	36	8.3445 (i)	1.344	na	8.3837 (0,47%)
C10	40	52.0535	2.891	na	90.696 (74%)

Table 2: Costs and running times

In Table 2 the results are shown; Table 3 contains the corresponding satellite assignment. The first column in Table 2 contains the name of the instance, and the second column gives the number of satellites (n). The third column gives the value of an optimal solution, and indicates (by "(i)") whether the linear programming (LP) relaxation of model (4)-(8) for the corresponding instance is integral. The fourth column gives the running time (in seconds) needed to solve the integer program (4)-(8). Columns 5 and 6 each give the value of a feasible solution: Column 5 reports the outcome of a local search procedure (called E-P2P) described in Dutta and Tsiotras [2], and Column 6 (called "restricted") gives the best possible solution when moving satellites need to return to their original position.

Table 2 shows that the software finds an optimal solution within 1 second for the instances with size 18 or smaller, and in a few seconds for instances with size 40 or less. The LP relaxation yields an optimal solution for all but one instance tested. The optimal solutions reported here show that the solutions found by local search are quite good: with one exception (instance C8), the optimal solution is only 1-3 % better than the solution found by local search. When we compare the optimal solutions to the solutions with the added restriction that satellites need to return to their original position (which can be found in polynomial time), it turns out the improvement is often significant.

5 Conclusion

In this note we show that solving the peer-to-peer refueling problem to optimality is a computationally hard problem. However, since instances from practice contain 50 satellites or less, finding the solution with minimum fuel costs can be done using state-of-the-art integer programming software.

References

[1] Burkard, R.E., R. Rudolf, and G.J. Woeginger (1996), *Three-dimensional axial assignment problems with decomposable costs*, Discrete Applied Mathematics **65**, 123-139.

[2] Dutta, A. and P. Tsiotras (2008), *Egalitarian Peer-to-Peer Satellite Refueling Strategy*, Journal of Spacecraft and Rockets **45**, 608-618.

[3] Garey, M.R. and D.S. Johnson (1979), Computers and intractability: A guide to the theory of NP-completeness, *W.H. Freeman and Co.*, New York.

[4] Schrijver, A. (2003), Combinatorial Optimization: Polyhedra and Efficiency, *Springer*, Berlin.

[5] Spieksma, F.C.R. (2000), *Multi Index Assignment Problems: Complexity, Approximation, Applications*, in: Nonlinear Assignment Problems, edited by P. Pardalos and L. Pitsoulis, Kluwer, Dordrecht.

Constellation	Assignment
C1	$s_1 \rightarrow s_4 \rightarrow s_5, s_3 \rightarrow s_2 \rightarrow s_3, s_5 \rightarrow s_9 \rightarrow s_8, s_6 \rightarrow s_{10} \rightarrow s_1, s_8 \rightarrow s_7 \rightarrow s_6$
C2	$s_1 \rightarrow s_{12} \rightarrow s_{11}, s_4 \rightarrow s_9 \rightarrow s_{10}, s_6 \rightarrow s_7 \rightarrow s_8, s_8 \rightarrow s_5 \rightarrow s_6, s_{10} \rightarrow s_3 \rightarrow s_4,$ $s_{11} \rightarrow s_{16} \rightarrow s_{15}, s_{13} \rightarrow s_2 \rightarrow s_1, s_{15} \rightarrow s_{14} \rightarrow s_{13}$
C3	$s_2 \rightarrow s_5 \rightarrow s_6, s_4 \rightarrow s_3 \rightarrow s_4, s_6 \rightarrow s_1 \rightarrow s_2, s_7 \rightarrow s_{10} \rightarrow s_9, s_9 \rightarrow s_{12} \rightarrow s_{11},$ $s_{11} \rightarrow s_8 \rightarrow s_7$
C4	$s_2 \rightarrow s_{17} \rightarrow s_{16}, s_4 \rightarrow s_{15} \rightarrow s_{14}, s_7 \rightarrow s_{10} \rightarrow s_{11}, s_9 \rightarrow s_{12} \rightarrow s_{13}, s_{11} \rightarrow s_8 \rightarrow s_9,$ $s_{13} \rightarrow s_6 \rightarrow s_7, s_{14} \rightarrow s_5 \rightarrow s_4, s_{16} \rightarrow s_1 \rightarrow s_{18}, s_{18} \rightarrow s_3 \rightarrow s_2$
C5	$s_1 \rightarrow s_{12} \rightarrow s_{11}, s_4 \rightarrow s_7 \rightarrow s_8, s_6 \rightarrow s_9 \rightarrow s_{10}, s_8 \rightarrow s_5 \rightarrow s_6, s_{10} \rightarrow s_2 \rightarrow s_1, s_{11} \rightarrow s_3 \rightarrow s_4$
C6	$s_2 \rightarrow s_{13} \rightarrow s_{12}, s_4 \rightarrow s_{11} \rightarrow s_{10}, s_7 \rightarrow s_8 \rightarrow s_9, s_9 \rightarrow s_6 \rightarrow s_7, s_{10} \rightarrow s_5 \rightarrow s_4,$ $s_{12} \rightarrow s_1 \rightarrow s_{14}, s_{14} \rightarrow s_3 \rightarrow s_2$
C7	$s_1 \rightarrow s_{16} \rightarrow s_1, s_3 \rightarrow s_{14} \rightarrow s_{15}, s_4 \rightarrow s_{13} \rightarrow s_{12}, s_6 \rightarrow s_9 \rightarrow s_8, s_8 \rightarrow s_{11} \rightarrow s_{10},$ $s_{10} \rightarrow s_7 \rightarrow s_6, s_{12} \rightarrow s_5 \rightarrow s_4, s_{15} \rightarrow s_2 \rightarrow s_3$
C8	$s_1 \rightarrow s_2 \rightarrow s_3, s_3 \rightarrow s_4 \rightarrow s_5, s_5 \rightarrow s_6 \rightarrow s_7, s_7 \rightarrow s_8 \rightarrow s_9, s_9 \rightarrow s_{10} \rightarrow s_{11},$ $s_{11} \rightarrow s_{12} \rightarrow s_{13}, s_{13} \rightarrow s_{14} \rightarrow s_{15}, s_{15} \rightarrow s_{16} \rightarrow s_1$
C9	$s_1 \rightarrow s_2 \rightarrow s_3, s_3 \rightarrow s_4 \rightarrow s_5, s_5 \rightarrow s_6 \rightarrow s_7, s_7 \rightarrow s_8 \rightarrow s_9, s_9 \rightarrow s_{10} \rightarrow s_{11},$ $s_{11} \rightarrow s_{12} \rightarrow s_{13}, s_{13} \rightarrow s_{14} \rightarrow s_{15}, s_{15} \rightarrow s_{16} \rightarrow s_{17}, s_{17} \rightarrow s_{18} \rightarrow s_{19},$ $s_{19} \rightarrow s_{20} \rightarrow s_{21}, s_{21} \rightarrow s_{22} \rightarrow s_{23}, s_{23} \rightarrow s_{24} \rightarrow s_{25}, s_{25} \rightarrow s_{26} \rightarrow s_{27},$ $s_{27} \rightarrow s_{28} \rightarrow s_{29}, s_{29} \rightarrow s_{30} \rightarrow s_{31}, s_{31} \rightarrow s_{32} \rightarrow s_{33}, s_{33} \rightarrow s_{34} \rightarrow s_{35},$ $s_{35} \rightarrow s_{36} \rightarrow s_1$
C10	$s_1 \rightarrow s_{40} \rightarrow s_{39}, s_3 \rightarrow s_{38} \rightarrow s_{37}, s_5 \rightarrow s_{36} \rightarrow s_{35}, s_7 \rightarrow s_{34} \rightarrow s_{33}, s_9 \rightarrow s_{32} \rightarrow s_{31},$ $s_{12} \rightarrow s_{21} \rightarrow s_{22}, s_{14} \rightarrow s_{23} \rightarrow s_{24}, s_{16} \rightarrow s_{25} \rightarrow s_{26}, s_{18} \rightarrow s_{27} \rightarrow s_{28},$ $s_{20} \rightarrow s_{29} \rightarrow s_{30}, s_{22} \rightarrow s_{19} \rightarrow s_{20}, s_{24} \rightarrow s_{17} \rightarrow s_{18}, s_{26} \rightarrow s_{15} \rightarrow s_{16},$ $s_{28} \rightarrow s_{13} \rightarrow s_{14}, s_{30} \rightarrow s_{11} \rightarrow s_{12}, s_{31} \rightarrow s_2 \rightarrow s_1, s_{33} \rightarrow s_4 \rightarrow s_3,$ $s_{35} \rightarrow s_6 \rightarrow s_5, s_{37} \rightarrow s_8 \rightarrow s_7, s_{39} \rightarrow s_{10} \rightarrow s_9$

Table 3: Satellite assignment