Satellite Payload Reconfiguration: an ILP Model

Apostolos Stathakis\textsuperscript{1}, Grégoire Danoy\textsuperscript{2}, Pascal Bouvry\textsuperscript{2}, and Gianluigi Morelli\textsuperscript{3}

\textsuperscript{1} Interdisciplinary Centre for Security, Reliability, and Trust, University of Luxembourg
apostolos.stathakis@uni.lu
\textsuperscript{2} CSC Research Unit, University of Luxembourg
gregoire.danoy@uni.lu, pascal.bouvry@uni.lu
\textsuperscript{3} SES Engineering, Betzdorf, Luxembourg
gianluigi.morelli@ses.com

Abstract. The increasing size and complexity of communication satellites has made the manual management of their payloads by engineers through computerised schematics difficult and error prone. This article proposes to optimise payload reconfigurations for current and next generation satellites using a novel Integer Linear Programming model (ILP), which is a variant of network flow models. Experimental results using CPLEX demonstrate the efficiency and scalability of the approach up to realistic satellite payloads sizes and configurations.

Keywords: Integer-Linear programming, multicommodity flow, satellite payload

1 Introduction

With an expected lifetime of 15 years or more, telecommunication satellites require flexibility and efficiency to answer potential changing service requirements or hardware failures during that period. To meet this demand, the complexity of satellites’ payloads, which play the main role in signal transmissions, has increased significantly. A payload is composed of a number of hardware components, such as amplifiers and channel filters. In addition, a large number of switches, interconnected to compose switch matrices, are used to ensure signal routings among payload components (selectivity) and functionality in case of failures (redundancy). Currently, payload engineers use computerised schematics to graphically represent the payload structures and find workable configurations. However, finding optimal payload reconfigurations and signal re-routing solutions has become a hard task due to the increasing complexity of the switch matrices, making this manual management of the payload difficult and time consuming. Commercial software solutions exist that have built-in optimisers \cite{4,7}, however, as they are closed packages, they do not provide the advantage of the flexibility achieved with computerised schematics.

The work proposed in this paper will help payload engineers to configure current and future satellites payloads via a novel optimisation tool that works...
in conjunction with the existing computerised schematics. More precisely, we here propose to solve the payload configuration process using an integer-linear programming (ILP) model, i.e. a variant of network flow problems, which to the best of our knowledge has never been tackled in the literature. This mathematical model should be flexible and scalable respectively to different types and sizes of real satellite payloads.

The remainder of this paper is organized as follows. The next section provides a state of the art analysis on payload reconfiguration softwares and network flow problems. Section 3 presents a detailed description of the payload reconfiguration problem and a description of the proposed mathematical model. Experimental results using the ILOG CPLEX solver are presented in Section 4. Finally, section 5 concludes our work and provides some perspectives.

2 State of the Art

The main function of the satellite payload system is to receive the uplink signals from the ground stations, to route those signals to apply frequency conversion and amplification, and to finally retransmit those modified signals on the downlink [8]. The signals’ routing flexibility is currently achieved through reconfigurable payload components, i.e. different switch types organized in matrices. These permit to modify the signals routings in order to meet new operational needs. As an example, in case of failure of an amplifier, the signals are rerouted to a spare one. This reconfiguration process is initiated through telecommands, which are submitted from the ground stations, by the payload engineers. However this flexibility comes to the expense of a higher payload size and complexity. Indeed, finding an optimal reconfiguration that satisfies operational objectives has become a time consuming and error prone task for engineers.

Two commercial softwares dedicated to payload reconfiguration optimisation exist. Smart Rings [4] uses a recursive search combined with automated deduction to compute all possible payload reconfiguration solutions and provides information on each solution quality. TRECS [7] is a second software package that allows the user to define the payload model per satellite and proposes an Auto-Solve feature that can plan a reconfiguration to limit or minimize the number of necessary changes. However, payload engineers would benefit from an open interaction between the computerised schematics they use to describe the payload models and the optimisation tools, rather than adapting to new interface environments. Besides, these tools act as ‘black boxes’, neither allowing flexibility nor knowledge extraction concerning the algorithms used. Finally, they lack of openness since their adaptation, e.g. in case of newer payload components, might be difficult or time consuming.

In this paper, an original Integer Linear Programming (ILP) model, based on network models, is proposed for solving the satellite payload reconfiguration problem. Network optimisation problems is a well studied area [1], which consist of supply and demand points, together with several routes that connect these points and are used to transfer the supply to the demand [3]. Network problems
such as shortest path, assignment, max-flow, transshipment, vehicle routing, and multicommodity flow problems, constitute the most common classes of practical optimisation problems [3]. Such models are widely used in telecommunications and satellite networks for optimal routing [2] and network design [6, 5].

In our approach, we consider electromagnetic signals (flows) crossing the payload network structure to reach the suitable outputs. However, a direct application of network flow models is not possible due to some payload properties. Indeed each solution to the problem, that consists in a set of switches positions, will define a different network topology. In order to handle these problem properties, e.g. types of switches or priority on switches to move, and considering the requirements for full control by the payload engineers, we propose an ILP model of the satellite payload reconfiguration problem that to the best of our knowledge has not been tackled in the literature.

3 Satellite Payload Reconfiguration: an ILP model

3.1 Problem description

Switch matrices are used in satellite payloads to provide system redundancy and to enhance the satellite capacity by providing full and flexible interconnectivity between received and transmitted signals. Through telecommands, the state of switches can be changed and each state defines different connectivities, as shown in Figure 2. Through such connectivities, the input channel signals cross the input switch matrix and are propagated to the appropriate amplifiers. Signals are then routed through the output switch matrix to the corresponding outputs. The objective of this optimisation problem is therefore to minimize the required number of switch changes to reconfigure the payload, which will intrinsically limit the risk of switch failures.

A simple problem instance with 32 switches of C-type (2 states) and R-type (4 states), is shown in Figure 1. In this example, the input channels 1 and 3 are activated and need to be connected to the corresponding output channels 1 and 3. The proposed solution requires 11 switch changes (shown in darker color) and activates amplifiers 2 and 3. In this work we tackled small to medium switch matrices sizes which correspond to realistic satellite payloads and investigated the problem of finding optimal configurations for establishing an initial set of connectivities.

3.2 The Mathematical Model

In our proposed model the signals are flowing through the network and each signal is assigned a distinct integer flow value. These values are propagated through the network according to flow propagation constraints. The capacity on each link is equal to one, i.e., a physical link can not be shared by different signals. Initially and for simplicity we consider only amplifiers, channels and switch matrices as payload components.
Fig. 1. A small problem instance with C and R switch types with a solution for connecting channels 1 and 3 with 11 switch changes, using amplifiers 2 and 3.

Constants and Sets:

Let define:
- \( q \) as the number of channels to be connected.
- \( n \) as the maximum number of states among all switches of the network (e.g. if R switches with 4 states and C switches with 2 states are used, \( n=4 \)).
- \( P \) as a set of size \( n \), with integer values from 0 to \( n-1 \), representing the maximum number of positions a switch can have. The position of a switch refers to the number of steps the switch should take from its current state to reach a new state.
- \( S \) as the set of all switches.
- \( T \) as the set of all amplifiers.
- \( C \) as the set of all channels.
- \( L \) as the set of all links. \( L \) contains the following subsets:
  - set \( L = CL \cup TL \cup SL \) with:
    - \( CL \) as the set of all links connected to the channel filters, \( CL = CL_{in} \cup CL_{out} \), where \( CL_{in} \) of size \( |C| \), the set of all links neighboring with the channel filters on the input and \( CL_{out} \) of size \( |C| \), the set of all links neighboring with the channel filters on the output. Let \( cl_{in} \) be the link connected with channel \( c \) on the input and \( cl_{out} \) be the link connected with channel \( c \) on the output.
    - \( TL \) as the set of all links connected to amplifiers. \( TL = TL_{in} \cup TL_{out} \), with \( TL_{in} \) of size \( |T| \), the set of input links of all amplifiers and \( TL_{out} \) of size \( |T| \), the set of output links of all amplifiers. Let \( tl_{in} \) be the input link of amplifier \( t \) and \( tl_{out} \) be the output link of amplifier \( t \).
    - \( SL \) as the set of all links between any two switches.
  - \( l_0 \) as a special link, or link 0, when signal can not be propagated. For instance in Figure 2, with a switch in state 2, link \( a \) is connected with \( l_0 \).
  - \( CL_{conn} \subseteq CL_{in} \) of size \( q \), as the set of links neighboring with the input channels that will be connected.
We define $M$ a matrix of size $|S| \times |P| \times |L| \times |L \cup \{ l_0 \}|$ with:

$$m_{s,p,i,j} = \begin{cases} 1 & \text{if } l_i \text{ is connected with } l_j \text{ via switch } s \text{ at position } p, \\ 0 & \text{otherwise}. \end{cases}$$

**Variables**

- Let define $Pos$ of size $|S|$, as the solution vector, providing the position for each switch. For instance, if an R-type switch (with 4 states) is initially at state 3, and needs to be in $pos_s = 1$, it will have to move to state 4.
- Binary vector $Change$ of size $|S|$, indicating whether a switch needs to change its initial state or not.
  $$change_s = \begin{cases} 1 & \text{if } pos_s > 0, \\ 0 & \text{otherwise}. \end{cases}$$
- Integer vector $Flow$ of size $|L|$, showing the flow value (from 0 to q) that is carried by each link. It follows:
  $$flow_l = \{ x \text{ with } 0 < x \leq q, x \in \mathbb{Z} \text{ if link } l \text{ is used}, \\ 0 \text{ otherwise}. \}$$
- Binary vector $B$ of size $|S| \times |P|$, is used to activate or de-active the flow propagation constraints, such that:
  $$b_{s,p} = \begin{cases} 1 & \text{if } pos_s = p, \\ 0 & \text{otherwise}. \end{cases}$$
- boolean vector $Ampused$ of size $|T|$, indicating whether an amplifier is active (used) or not.

**Constraints**

- To start the flow distribution, we assign positive distinct flow values to each link of set $CL_{conn}$:
  $$flow_{l_i} = k_i, 0 < k_i \leq q, k_i \in \mathbb{Z}, \forall l_i \in CL_{conn}.$$
- To avoid flow paths starting from an input channel and returning in another input channel, flow values must be set to 0 for all unused input channels:
  $$flow_{l_i} = 0, \forall l_i \in \{ CL_m \} - \{ CL_{conn} \}.$$
- We ensure that each input channel signal reaches the corresponding output:
  $$flow_{cl_{inc}} = flow_{cl_{outc}}, \forall c \in C.$$
The flow propagation constraints are expressed for each switch and describe the established connections at each switch state. We ensure that connected links have the same flow value propagated. Figure 2 shows the non-linear expressions for flow propagation constraints for all states of an R-type switch. The linear equivalent constraints are expressed as follows:

\begin{align*}
flow_{l_1} + b_{s,p} \cdot q - q &\leq \flow_{l_2} \leq \flow_{l_1} - b_{s,p} \cdot q + q : \\
\forall s \in S, \forall p \in P, \forall l_1, l_2 \in (L \cup \{l_0\})^2, \text{ such as } m_{s,p,l_1,l_2} = 1.
\end{align*}

If \(b_{s,p} = 0\) the constraint is deactivated. If \(b_{s,p} = 1\) the flow value will be propagated. For the C-type switches that have only 2 available states we set \(b_{s,2} = b_{s,3} = 0\). The flow propagation constraints for the initial state of each switch are described for \(b_{s,0} = 1\).

- We also ensure that only one switch position is selected each time:

\[ \sum_{p \in P} b_{s,p} = 1, \quad \forall s \in S. \]

\[ \pos_s = \sum_{p \in P} p \cdot b_{s,p}, \quad \forall s \in S. \]

- We ensure that \(\change_s = 1\), if and only if \(\pos_s > 0\):

\[ \change_s \cdot n \geq \pos_s, \quad \forall s \in S. \]

- We ensure the flow propagation through an amplifier and that the number of active amplifiers equals the number of connected channels:

\[ \flow_{tl_{int}} = \flow_{tl_{out}}, \quad \forall t \in T. \]

\[ \ampused_t \cdot q \geq \flow_{tl_{int}}, \quad \forall t \in T. \]

\[ \sum_{t \in T} \ampused_t = q. \]
Link \{l_0\} never carries any signal.

\[ flow_{l_0} = 0. \]

**Objectives**

\[ \text{Min} \sum_{s \in S} \text{change}_s. \]

**Complete ILP model**

- **Variables**:
  - \( pos_s \in \mathbb{Z} \) \( \forall s \in S \)
  - \( change_s \in \{0; 1\} \) \( \forall s \in S \)
  - \( flow_l \in \mathbb{Z} \) \( \forall l \in L \cup \{l_0\} \)
  - \( b_{s,p} \in \{0; 1\} \) \( \forall s \in S, \forall p \in P \)
  - \( \text{ampused}_t \in \{0; 1\} \) \( \forall t \in T \)

- **Objective Function**:
  - \( \text{Min} \sum_{s \in S} \text{change}_s \)

- **Constraints**:
  - \( flow_{l_1} = k_i \) \( 0 < k_i \leq q, k_i \in \mathbb{Z}, \forall l_i \in CL_{\text{conn}} \)
  - \( flow_{l_1} = 0 \) \( \forall l_i \in \{CL_{\text{in}}\} - \{CL_{\text{conn}}\} \)
  - \( flow_{c_{\text{in}}} = flow_{c_{\text{out}}} \) \( \forall c \in C \)
  - \( flow_{l_1} + b_{s,p} \cdot q - q \leq flow_{l_2} \leq flow_{l_1} - b_{s,p} \cdot q + q \) \( \forall s \in S, \forall p \in P, \forall (l_1, l_2) \in (L \cup \{l_0\})^2, \text{s.t. } m_{s,p,l_1,l_2} = 1. \)
  - \( \sum_{p \in P} b_{s,p} = 1 \) \( \forall s \in S \)
  - \( pos_s = \sum_{p \in P} p \cdot b_{s,p} \) \( \forall s \in S \)
  - \( change_s \cdot n \geq pos_s \) \( \forall s \in S \)
  - \( flow_{l_1,\text{in}} = flow_{l_1,\text{out}} \) \( \forall t \in T \)
  - \( \sum_{t \in T} \text{ampused}_t \cdot q \geq flow_{l_1,\text{in}} \) \( \forall t \in T \)
  - \( flow_{l_0} = 0 \)

**3.3 Flexibility of the Model**

The proposed model permits to set backup (or failed) amplifier(s) by setting their flow value to 0. A switch may also fail in a usable or non-usable state, which respectively implies fixing its position or removing it from the model. Due to frequency constraints, not all amplifiers are appropriate for all channels. It is possible to define which amplifier to connect on which channel by assigning the same flow value (or a range of values for subset of channels) to the amplifier links.
Table 1. Experimental results on different payloads configurations

<table>
<thead>
<tr>
<th>Number of switches</th>
<th>Switches type</th>
<th>Number of amplifiers</th>
<th>Number of required connections</th>
<th>Average number of switch changes</th>
<th>Average Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>R</td>
<td>10</td>
<td>5</td>
<td>4 ± 2.17</td>
<td>0.206 ± 0.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>6.6 ± 1.52</td>
<td>0.447 ± 0.47</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>9.6 ± 2.61</td>
<td>0.694 ± 0.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>9.4 ± 2.30</td>
<td>1.914 ± 1.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>11 ± 3.70</td>
<td>2.478 ± 1.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>13 ± 3.51</td>
<td>2.212 ± 2.46</td>
</tr>
<tr>
<td>50</td>
<td>R</td>
<td>10</td>
<td>5</td>
<td>9 ± 3.67</td>
<td>1.975 ± 1.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>9.8 ± 3.27</td>
<td>4.532 ± 3.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>10 ± 3.19</td>
<td>4.214 ± 7.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>14.8 ± 1.93</td>
<td>16.231 ± 7.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>17 ± 3.94</td>
<td>166.611 ± 83.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>20 ± 3.17</td>
<td>310.151 ± 183.86</td>
</tr>
<tr>
<td>90</td>
<td>R</td>
<td>30</td>
<td>15</td>
<td>11 ± 3.10</td>
<td>4.322 ± 3.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>10.8 ± 4.55</td>
<td>217.003 ± 466.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td>13 ± 4.27</td>
<td>818.926 ± 1698.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>19.4 ± 2.54</td>
<td>8329.019 ± 33443.70</td>
</tr>
<tr>
<td>100</td>
<td>T,C,R</td>
<td>30</td>
<td>15</td>
<td>9 ± 3.16</td>
<td>1197.784 ± 35523.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>9.8 ± 2.86</td>
<td>44.664 ± 47.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td>14 ± 3.78</td>
<td>480.440 ± 667.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>14 ± 2.77</td>
<td>31 ± 304 ± 277.35</td>
</tr>
</tbody>
</table>

4 Experimental Results

The proposed model was implemented in AMPL and experiments were conducted on different payload sizes using an ILP solver with a time limit: CPLEX 11 on a machine with two Intel Xeon X5355 2.66 GHz processors with 4 cores each, 32GB of RAM and running Ubuntu 8.04.

Experiments have been conducted on 4 groups of random payload instances, each one described in Table 1 by the number and type of switches used and the number of usable amplifiers. Each instance consists in connecting 5 sets of randomly chosen channels with no pre-existing connection (see column ‘Number of required connections’). This will permit to analyze the influence of the subset of channels to connect on the solution quality and computational time. No pre-existing connection represents ‘worst case’ problems, since reconfiguring payloads with some pre-established connections would imply smaller search spaces. The last two columns respectively provide the average value and standard deviation of the objective function (number of switch changes) and the average required computational time for solving our problem instances.

Table 1 provides a scalability study of the proposed approach. All solutions have been validated using the computerised schematics. In terms of solutions quality, the number of switch changes increases as the number of required connections increases, in all but one case (90 switches, 30 amplifiers and 20 channels to connect). It can be noticed that the number of switch changes is lower for all the 100 switch instances that the 90 switch ones. The solution quality thus
depends on the topology of the switch network and on the type of switch used. Indeed in the 100 switch instance most of the switches are of type T and C, which ensure all the necessary connectivities with respectively only 3 and 2 possible states. The important variance in the number of switch changes also indicates that the chosen set of channels to be connected has an important impact on the solution.

In terms of computational time taken by CPLEX, it generally also increases with the number of switches that compose the switch matrix, and the number of required connectivities. The exceptions are with 40 switches and 10 channels; 50 switches and 7 channels; and with 100 switches. As for the solution quality, this can be explained by the topology and type of switches of the payload. The variance in the computational time the deviation value is very high, especially for our large size payload systems (90 and 100 switches with 30 amplifiers). The chosen set of channels to connect has thus a key influence on the result. One possible explanation is that some channels may require less switch crossings to reach a set of amplifiers. To investigate this impact, it is interesting to compute for any channel the number of switch crossings that is required to connect to any amplifier. This will be investigated in our future work.

5 Conclusions and Future Work

In this paper, we proposed to tackle for the first time the optimisation of satellites’ payload reconfiguration with an ILP model, based on network flow models. Experimental results using the CPLEX solver have been validated using computerised schematics and the scalability of the model has been studied on payloads of different sizes, up to realistic ones (from 40 to 100 switches).

As future work, we plan to further enhance the model in order to meet new operational objectives. Among them, the minimization of the path losses and number of signals interruptions is a direct next step. The robustness of the solution will also be considered, in order to ensure the functionality of the system in cases of potential failures. Finally, after determining new objectives, we will consider the definition and optimisation of multi-objective versions of the payload reconfiguration problem.

References

7. TRECS: Transponder reconfiguration system [online], http://www.integ.com/TRECS.html