Hybridisation Schemes for Communication Satellite Payload Configuration Optimisation

Apostolos Stathakis[a⋆], Grégoire Danoy[b], El-Ghazali Talbi[c], Pascal Bouvry[b] and Gianluigi Morelli[d]

[a] *Interdisciplinary Centre for Security, Reliability, and Trust, University of Luxembourg*;
[b] *CSC Research Unit, University of Luxembourg*;
[c] *INRIA-Lille Nord Europe, Université Lille 1, France*;
[d] *SES Engineering, Betzdorf, Luxembourg*

Abstract. *The increasing complexity of current telecommunication satellite payloads has made their manual management a difficult and error prone task. As a consequence, efficient optimisation techniques are required to help engineers to configure the payload. Recent works focusing on exact approaches faced scalability issues while metaheuristics provided unsatisfactory solution quality. This work therefore proposes three hybridisation schemes that combine both metaheuristics and an exact method. We focus on the initial configuration problem case and we consider as objective to minimise the length of the longest channel path. Experimental results on realistic payload sizes demonstrate the advantage of those approaches in terms of efficiency within a strict operational time constraint of ten minutes on a single CPU core.*

Keywords: satellite payload configuration, optimisation, hybrid metaheuristics

1 Introduction

The communication satellite consists of the payload and the platform. The payload, which plays the main role in the transmission of the signals, includes all the necessary electronic equipment like multiplexers, switches and amplifiers, as well as the receiving and transmitting antennas. The platform consists of all the subsystems that allow the payload to function, like the electric power supply. In the payload the routing of the signals is achieved through reconfigurable switches organised in switch matrices. But the modern operational requirements and the increasing demands of the market have led to large and complex payloads. Numerous amplifiers and large switch matrices are used to ensure flexibility in signal routings and functionality in case of failures (redundancy). As a consequence, the current manual management of the payload, with the use of computerised

⋆ Corresponding author. Email: apostolos.stathakis@uni.lu

schematics, is becoming hard and time consuming. Commercial software solutions exist for payload configuration that have built-in optimisers, but as they are closed packages, they do not provide the advantage of the flexibility achieved with the computerised schematics.

To deal with this challenging problem, Stathakis et. al. have proposed an integer linear programming (ILP) model aiming to find solutions using exact methods [8][10]. It was demonstrated though that some instances could not be solved exactly within a time constraint of ten minutes defined by engineers. To overpass this limitation the same authors proposed to apply genetic algorithms for the first time to the considered problem [9] and more instances were solved successfully within the time constraint, with a difference though in the quality of the obtained solutions compared to the optimal ones.

In this article we propose three hybridisation schemes that integrate a local search, a cellular genetic algorithm and the ILP based exact method. We target to solve the real size problem instances within the strict constraint of ten minutes. The experimental results denote the efficiency of the proposed approaches when tackling large payload instances, in both computational time and fitness values.

The remainder of this paper is organized as follows. The next section presents the related works in the subject. Section 3 addresses the considered problem in more details and in Section 4 the proposed hybridisation schemes are detailed. The experimental results are analysed in Section 5 and finally the conclusions and some perspectives for future work are provided.

2 Related Works

In order to meet the modern demands of the market and to achieve the current operational requirements, the satellite industry needs to reach several levels of flexibility as they have been described in [4]. They concern among others flexibility in power allocation (assigning various power levels to channels to cope with traffic allocations), in coverage definition (shape of coverage, number and size of beams in case of multi-beam coverage) and at the channel routing level that we are focusing on. To ensure such flexibility the current payloads are composed with numerous amplifiers and large switch matrices.

The design of the optimal switch matrix topology that will satisfy the given operational requirements is the first optimisation problem which is faced by satellite manufacturers, when constructing the spacecrafts. Switches are expensive components and thus designing a topology that will satisfy all the routing requirements, given a number of channels and amplifiers, while minimising the cost, is of importance. Research works have tackled this design problem, like in [3] where a graph-based analysis is used to minimise the number of needed switches, while ensuring a fault-tolerant switch matrix design.

From the satellite operator point of view, which is of our interest, efficient techniques are also required to find optimal configurations when new operational and business demands arise. Commercial software packages exist, like TRECS [12] and Smartrings [6]. Details concerning the algorithms and the mod-

els used by both packages are not accessible due to commercial restrictions. Smartrings applies a recursive search to compute all possible payload configurations, while the algorithm is controlled by constraints like the number of switches used or the number of the interrupted channel paths. Similarly, TRECS uses an algorithm to find all feasible solutions and sorts them by output signal quality, while rejecting many millions of non-feasible solutions. However those packages lack flexibility. Their closed APIs do not allow efficient interaction and integrations in company workflows. Besides, they use a black-box solver that can not be changed or customised based on the problem that has to be solved, which would allow the application of different single or multi-objective algorithms.

On the academic side, few works have been proposed that deal with the considered problem. In [7] a recursive algorithm was proposed to perform a breadth-first-search (BFS) in order to find all feasible paths that connect channels to amplifiers. Experiments showed the efficiency of the proposed method on a small switch network. However, it can be expected that for larger problems the BFS algorithm will be limited due to its time complexity as every vertex and every edge will be explored in the worst case.

Stathakis et al. [9] proposed and validated an Integer Linear Programming (ILP) optimisation model that can be used for applying single and multi objective algorithms. The model allows the optimisation of specific objectives like the number of switch changes. Instead of enumerating all feasible solutions, the authors aim at a single optimal solution or a set of non-dominated solutions. The same authors extended their mathematical model to minimise the channel interruptions for the reconfiguration problem [10]. As has been demonstrated, not all instances could be solved exactly within the ten minutes constraint. To overcome this limitation metaheuristics were applied for the first time to the considered problem [9], which permitted to solve more instances with a degradation though between their fitness and the one found by the exact.

In this work, we propose novel hybridisation schemes for the considered problem that integrate a local search with a cellular genetic algorithm and the ILP based exact method. The experimental results confirmed the efficiency of those schemes, and thus can be used when tackling even larger payload problem sizes. The proposed methods are high-level relay hybrids according to the taxonomy of hybrid algorithms used in [11].

3 Problem Description

A simplified payload switch matrix example is shown in Fig.1, which is composed by 16 switches. The input channel signals are crossing the switch matrix and are guided for amplification. Different types of switches may be used and each one has different positions allowing different paths. The four possible positions of an R-type switch are shown in Fig.2, whereas the C-type switch has only 2 possible positions. A link (connector between any two payload components) can be used by only a single channel. The switch matrix can have any design topology (not symmetric necessarily) and channels can be placed at any location of the matrix.

The problem of payload configuration can be decomposed into three related subproblems. The first one is the initial configuration problem, which consists in finding an optimal configuration for connecting an initial set of channels in a payload without any pre-connected channels. The second case is the reconfiguration problem, which occurs when there exists a set of pre-connected channel paths that carry services and some additional channels have to be activated. The third one is the restoration problem, which arises when a set of channels is already connected and one or more failures occur to the amplifiers or to the switches. In this case, the channels affected by these failure(s) have to be rerouted through different paths.

In this work we are focusing on the initial configuration problem. More precisely, given a set of channels to connect, the initial payload configuration problem consists in finding the positions of the switches that permit to establish the path from each required channel to an amplifier. Among several important objectives that are of interest, we choose to minimise the length of the longest channel paths. Long paths imply high signal attenuation and they cause restrictions on future reconfiguration processes. In Fig.1 one solution example is shown for connecting channel 1 to amplifier 2 and channel 3 to amplifier 4, where channel 1 follows a path of 7 switch crossings and channel 3 follows a path with 6 switch crossings i.e number of switches used in the path.

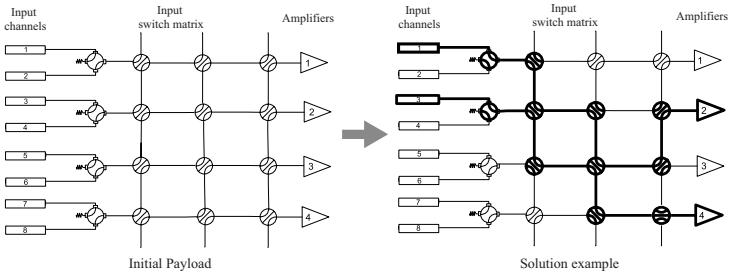

Fig. 1. Simplified initial payload instance and solution that connects channels 1 and 3 to amplifiers 2 and 4

More formally, let k be the number of switches in the switch network and the set C, of size n, the set of channels to connect. Let $path_c, c \in C$ be the length of the channel path c in switch crossings. The solution vector of size k: $P = \{pos_1, \ldots, pos_k\}$ denotes the position of each switch. The objective is to find a solution P which connects the n channels and minimises

$$max_{c \in C}(path_c) \tag{1}$$

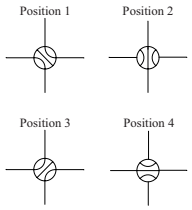

Fig. 2. Positions of one R-type switch

4 Proposed Methodology

An illustration of the three schemes with their components is provided in Fig.3. The first one, referred to as LSM, combines a local search with a cellular genetic algorithm. The rest two hybrids, referred to as LSMExB and LSMExP, combine LSM with the ILP based exact method. The local search and cellular genetic algorithm components are presented in the next two subsections. The ILP model is a variation of the one presented in [8]. In the last three subsections, the three hybrids are presented in details.

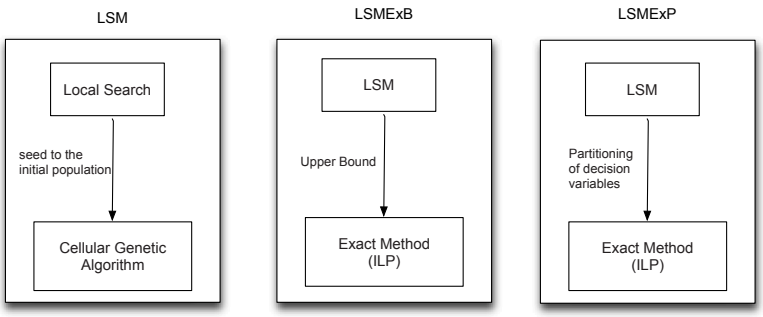

Fig. 3. The three hybrid schemes and their components

4.1 The Local Search Algorithm

The target of the local search algorithm is to find fast approximated payload configurations. The problem is regarded as a permutation problem. Since each path uses an available amplifier and defines positions on the switches, the ordering of channels to connect will influence the quality of the final solution. The next channels will be restricted in both the possible paths, and their final destination (amplifier). For n channels to connect, the solution of the local search algorithm is thus represented as a permutation of size n. We used a simple hill

climbing method where starting from a random solution (random permutation), at each iteration the current solution is replaced by a neighbor, using the first improvement strategy. The neighbors are generated with the use of the exchange operator. The local search terminates when there is no improvement. Each solution is evaluated using a greedy method to construct the paths.

A Greedy Method for Constructing Paths The payload can be represented as a graph $G = (V, E)$, where V is the set of payload components (switches, channels, amplifiers), and E the set of connectors (links) between them. We consider the length of each link to be equal to 1. Each component $u \in V$, has some coordinates (x, y) and at most 4 neighbor nodes (vertices) namely u_s, u_w, u_e, u_n, where s,w,e,n stand for south, west, east and north. The pseudocode of the greedy method used for finding the path of each channel is provided in Algorithm 1. Initially, the destination amplifier is chosen to be the closest available amplifier based on the Manhattan distance. We assume that any channel can be connected to any amplifier. If not, the destination will be chosen among the suitable amplifiers for each channel.

Algorithm 1 Pseudo-code of a greedy method to construct one channel path

```
 1: Input channel node c; path_c = {}
 2: dest = closest_to_c_available_amplifier
 3: current = switch_node_neighboring_to_c
 4: path_c.add(current)
 5: prec_node = channel node c
 6: while (current != dest) do
 7:     next_node = closest_to_dest_available_neighbor
 8:     if (next_node) then
 9:         updateNeighborhood(prec_node, current, next_node)
10:         prec_node = current
11:         current = next_node
12:         path_c.add(current)
13:     else
14:         path_c.clear
15:         break
16:     end if
17: end while
18: if (path_c is empty) then
19:     reset_All_neighbors()
20: else
21:     dest.available = false
22: end if
```

The next step is to set as current node the unique switch that is neighbor to the channel and the current node is added to the path (lines 3,4). As precedent node is set the channel node (line 5). To find the next node, the available

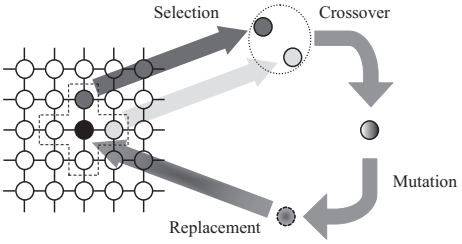

Fig. 4. cGA reproduction cycle with 5×5 population and L5 neighborhood.

neighbors of the current node are retrieved and the one which is closest to the destination is selected (line 7). Given the localities of the next and the precedent nodes compared to the current node, the position of the current switch is defined. The *updateNeighborhood* function which is then called (line 9) will update the available neighbors. In this function, the current switch defines which of its neighbors are available from now on. Besides, each of the neighbors of the current switch determines whether the current switch is considered as one of their available neighbors. The same process is repeated until the destination is reached. If the path can not be found, the neighbors of each switch are reset to the initial status (line 19). As long as a channel path is constructed, the used amplifier is set as not available (line 21).

4.2 The Cellular Genetic Algorithm

The cellular genetic algorithm (cGA) uses a single and structured population [2]. Individuals are spread in a two dimensional toroidal mesh and are only allowed to interact with their neighbors. An illustration of the cGA breeding loop is presented in Fig.4. The individuals are arranged on a 5X5 toroidal grid. The neighborhood of the center individual, linear 5, is presented as dashed lines. The representation of the solution is similar to the one used in [9], i.e. one individual represents the set of positions of all the switches. A binary encoding is used where a switch position is encoded using two bits. The binary vector is thus of size $2 * n$ with n the number of switches in the payload. Each solution describes a unique static graph. To assign a fitness value to the candidate solutions, we use the objective function:

$$F = r_c + \frac{lpl}{1000} \tag{2}$$

where r_c is the number of channels that have not been connected to an amplifier, from the set of n channels to connect, and lpl is the length of the longest found path, i.e. the number of switches used in this path.

4.3 The First Hybrid (LSM)

In the first hybrid, a seed individual provided by the local search is inserted in the initial population of cGA. The rest individuals are created uniformly at random. The global problem is solved by both methods. The two algorithms are applied in sequence. At first the problem is solved using local search and the solution is included to the initial population of the second metaheuristic. According to the taxonomy presented in [11], this hybrid is classified as High Level Relay Hybrid (HRH)(heterogenous, global, general) and is denoted as LSM.

4.4 The Second Hybrid (LSMExB)

The motivation for the second hybridisation scheme comes from the good performance of the cGA as demonstrated in [9]. The solution provided by the LSM can thus be considered as a good upper bound (length of the longest path) for the exact method. We applied this collaborative scheme aiming to improve the hit rate of the exact method (percentage of problems that were solved exactly within ten minutes). Thus, the upper bound of the objective function is set in the ILP model, based on the solution found by LSM and the exact method is then called. The global problem is solved by both methods. The two algorithms are applied in sequence. According to the taxonomy presented in [11], this hybrid is classified as High Level Relay Hybrid (HRH)(heterogenous, global, general) and is denoted as LSMExB.

4.5 Third Hybrid (LSMExP)

In this scheme we propose to partition the decision variables, as indicated in [11]. The decision variables can be partitioned in two sets X and Y. The variables of the set X will be fixed based on the solution found by LSM and the exact method will optimize the problem over the set Y. Hence, the generated problem is subject to free variables in the set Y and freesed variables in the set X. The decision variables that we fix are the positions of the switches based on some paths generated by the solution obtained by LSM. To summarise this hybrid, the global problem is solved by LSM and the partial problem is solved by the exact method. The two algorithms are applied in sequence. At first the problem is solved using metaheuristics, the values of some decision variables are set and the exact method is then called. According to the taxonomy presented in [11], this hybrid is classified as High Level Relay Hybrid (HRH)(heterogenous, partial, general), and is denoted as LSMExP.

5 Experimental Results

In this section we review the performances of the proposed schemes related to the hit rate (percentage of successfully solved instances), the fitness value and the required computational time.

5.1 Experimental Setup

All our experiments were carried out using the HPC facility of the University of Luxembourg, on a single CPU core of an Intel Xeon L5640 at 2.26GHz. The implementation of the metaheuristic algorithms was done using Paradiseo framework v.1.3 [5]. For the exact method CPLEX 12.4.0.0 was used [1].

Table 1. Parameters used for local search and cGA

LS	Algorithm	Hill Climbing
	Selection Strategy	First Improvment
	Neighbor operator	Exchange
cGA	Population	49, 7×7
	Selection	Binary tournament (BT), Current indiv. + BT
	Neighborhood	L5
	Crossover	DPX, p_c=0.8
	Mutation	bit flip, $p_m = \frac{1}{chrom\_length}$
	Replacement strategy	Replace if better
	Elitism	1 individual

For the local search and the cGA we used default parameters that are provided in Table 1. The problem instances tackled are the same as in [9], i.e switch matrix with 50 switches and and 23 amplifiers (maximum channels to connect). 30 channel instances of size 8, 13, 18 and 23 were randomly selected to connect. For the LSMExP we selected to fix the positions of the switches used by 4, 7, 10 and 12 randomly selected paths respectively.

5.2 Numerical Results

The results obtained after applying the ILP based exact method are displayed in Table 2. The hit rate, which represents for the exact method the percentage of instances that were solved exactly within ten minutes, decreases significantly from 90%, when 8 channels are connected, to 13.333% for the cases of 13 channels to connect. When connecting 18 channels only 6.666% of the instances were solved and none instance for the case of 23 channels (maximum size on a switch matrix with 23 amplifiers).

Concerning the comparison of LS, cGA and LSM, the average fitness and the hit rate are provided in Table 3. As can be seen the hybrid LSM outperforms the other methods. In terms of hit rate, which in this case denotes the percentage of instances where valid solutions were found (all paths constructed), LSM and cGA have both 100% for the instances of 8, 13 and 18 channels, but LSM has hit rate 82.333% when 23 channels are connected, compared to 80.888% for cGA. The hit rate of LS is 90%, 42.222%, 23.333% and 4.888% for 8, 13, 18 and 23 channels respectively. LSM provides also better fitness values, for example 0.00194 and

Table 2. Exact Method

Channels	Hit Rate(%)	Aver. Fitness	Aver. Time(sec)
8	90	$0.00181_{\pm 0.0004}$	$3.199_{\pm 12.896}$
13	13.333	$0.002_{\pm 0}$	$10.67_{\pm 13.647}$
18	6.666	$0.002_{\pm 0}$	$4.525_{\pm 4.122}$
23	0		

0.00304 when 8 and 13 channels are connected compared to 0.00195 and 0.00317 for cGA and for 18 and 23 channels LSM has average fitness 0.00365 and 0.18183 respectively compared to 0.00367 and 0.19879 for cGA. In terms of fitness, LSM outperforms cGA with statistical confidence for the case of 23 channels after performing the Wilcoxon test [13]. LSM has also smaller standard deviation in all cases except for the case of 18 channels. The fitness values of LS itself are 0.10346, 0.69998, 1.18435 and 2.84213 for 8, 13, 18 and 23 channels respectively.

Table 3. Fitness and HitRate(%)

#Ch	LS		cGA		LSM	
	fitness	hit-rate	fitness	hit-rate	fitness	hit-rate
8	$0.10346_{\pm 0.3}$	90	$0.00195_{\pm 0.0006}$	100.0	$0.00194_{\pm 0.0005}$	100
13	$0.69998_{\pm 0.69}$	42.222	$0.00317_{\pm 0.0006}$	100.0	$0.00304_{\pm 0.0005}$	100
18	$1.18435_{\pm 0.88}$	23.333	$0.00367_{\pm 0.0007}$	100.0	$0.00365_{\pm 0.0007}$	100
23	$2.84213_{\pm 1.33}$	4.888	$0.19879_{\pm 0.4}$	80.888	$0.18183_{\pm 0.381}$	82.333

The average convergence time of cGA and LSM, which denotes the average time needed by each method to reach the best solution found, is provided in Table 4. LSM converges faster in all cases. The highest difference occurs for 23 channels where LSM converged after 79.48sec and cGA in average after 91.85sec.

Table 4. Metaheuristics - Convergence Time(sec)

Channels	cGA	LSM
8	9.41	5.26
13	27.98	26.94
18	46.81	44.11
23	91.85	79.48

For the next hybrids LSMExB and LSMExP, LSM will run for the time provided in Table 4 and then the ILP based exact method is called so as the total termination condition is ten minutes in all experiments. The comparison

between LSMExB and LSMExP in terms of fitness and hitrate is presented in Table 5.

Table 5. Fitness and HitRate(%)

#Ch	LSMExB		LSMExP	
	fitness	hit-rate	fitness	hit-rate
8	$0.00181_{\pm 0.0004}$	90	$0.00191_{\pm 0.0005}$	97.888
13	$0.00260_{\pm 0.0005}$	16.666	$0.00300_{\pm 0.0005}$	93
18	$0.00200_{\pm 0}$	6.666	$0.00363_{\pm 0.0008}$	89.444
23	–	–	$0.00487_{\pm 0.0009}$	89.222

LSMExP method achieved 89.222% hit rate for the maximum case of connecting 23 channels compared to 0% for LSMExB. For 18 channels LSMExP solved 89.444% of instances compared to 6.666% of LSMExB. With LSMExP, the process is significantly accelerated since the exact method is optimising over a subset of the decision variables. LSMExB performed better compared to the ILP based exact method for the case of 13 channels as the hit rate of LSMExB is 16.666% compared to 13.333%. In Table 6 the average fitness is compared between LSM, LSMExB and LSMExP for the commonly solved instances. We observe that LSMExP provided the optimal solution for these instances (same fitness with LSMExB) and LSM provided solutions of less good quality.

Table 6. Fitness comparison on the commonly solved instances

Channels	LSMExB	LSM	LSMExP
8	$0.00181_{\pm 0.0004}$	$0.00182_{\pm 0.0003}$	$0.00181_{\pm 0.0004}$
13	$0.00260_{\pm 0.0005}$	$0.00261_{\pm 0.0005}$	$0.00260_{\pm 0.0005}$
18	$0.00200_{\pm 0}$	$0.00200_{\pm 0}$	$0.00200_{\pm 0}$

When connecting 8 channels, both LSMExB and LSMExP provided average fitness 0.00181 on the same instances compared to 0.00182 for LSM, and for 13 channels LSM has fitness 0.00261 compared to 0.00260 for the other two hybrids. For the 6.666% instances of 18 channels to connect the three methods had the same performance.

6 Conclusions and Perspectives

In this work we tackled the problem of optimal telecommunication satellite payload configuration. In order to further improve state-of-the-art results and solve the difficult problem instances within ten minutes on a single CPU core, we

proposed three hybridisation schemes. The first hybrid (LSM) is a high level relay hybrid that integrates a local search method with a cGA which improved the performance of the metaheuristic. The second hybrid (LSMExB) uses the solution of LSM as an upper bound for the exact method and permitted to solve more problem instances and in much shorter time. The last one (LSMExP) freezes a subset of decision variables based on the best solution obtained by LSM and uses the exact method to optimise the problem over the set of decision variables. With this last hybridisation method, the hitrate of solved instances was significantly increased. Future work will therefore focus on further improving this scheme, for instance by using a more intelligent way of choosing the number and the decision variables to fix. Its scalability will be experimented by using even larger payload instances.

References

1. Ibm ilog cplex. http://www.ilog.com/products/cplex/
2. Alba, E., Dorronsoro, B.: Cellular Genetic Algorithms. Operations Research/Compuer Science Interfaces, Springer-Verlag Heidelberg (2008)
3. Amini, O., Giroire, F., Prennes, S., Huc, F.: Minimal selectors and fault tolerant networks. Networks 55(4), 326–340 (2010), http://dx.doi.org/10.1002/net.20326
4. Balty, C., Gayrard, J.D., Agnieray, P.: Communication satellites to enter a new age of flexibility. Acta Astronautica 65(1-2), 75 – 81 (2009)
5. Cahon, S., Melab, N., Talbi, E.G.: Paradiseo: A framework for the reusable design of parallel and distributed metaheuristics. Journal of Heuristics 10(3), 357–380 (May 2004), http://dx.doi.org/10.1023/B:HEUR.0000026900.92269.ec
6. Chaumon, J., Gil, J., Beech, T., Garcia, G.: Smartrings: advanced tool for communications satellite payload reconfiguration. In: Aerospace Conference, 2006 IEEE. p. 11 pp. (2006)
7. Gulgonul, S., Koklukaya, E., Erturk, I., Tesneli, A.Y.: Communication satellite payload redundancy reconfiguration. In: Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on. pp. 1 –4 (oct 2012)
8. Stathakis, A., Danoy, G., Bouvry, P., Morelli, G.: Satellite payload reconfiguration optimisation: An ILP model. In: Pan, J.S., Chen, S.M., Nguyen, N.T. (eds.) ACIIDS (2). Lecture Notes in Computer Science, vol. 7197, pp. 311–320. Springer (2012)
9. Stathakis, A., Danoy, G., Schleich, J., Bouvry, P., Morelli, G.: Minimising longest path length in communication satellite payloads via metaheuristics. In: Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference. pp. 1365–1372. GECCO '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2463372.2463535
10. Stathakis, A., Danoy, G., Veneziano, T., Schleich, J., Bouvry, P.: Optimising satellite payload reconfiguration: An ILP approach for minimising channel interruptions. In: 2nd ESA Workshop on Advanced Flexible Telecom Payloads. pp. 1–8. European Space Agency (2012)
11. Talbi, E.G.: Metaheuristics - From Design to Implementation. Wiley (2009)
12. TRECS: Transponder reconfiguration system. http://www.integ.com/trecs.html
13. Wilcoxon, F.: Individual Comparisons by Ranking Methods. Biometrics Bulletin 1(6), 80–83 (1945)