



HAL
open science

HEURISTIC METHODS FOR TEST SEQUENCING IN TELECOMMUNICATION SATELLITES

Ludivine Boche-Sauvan, Bertrand Cabon, Marie-José Hugué, Emmanuel
Hébrard

► **To cite this version:**

Ludivine Boche-Sauvan, Bertrand Cabon, Marie-José Hugué, Emmanuel Hébrard. HEURISTIC METHODS FOR TEST SEQUENCING IN TELECOMMUNICATION SATELLITES . MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation, Nov 2014, Nancy, France. hal-01166683

HAL Id: hal-01166683

<https://hal.science/hal-01166683>

Submitted on 23 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HEURISTIC METHODS FOR TEST SEQUENCING IN TELECOMMUNICATION SATELLITES

L. BOCHE-SAUVAN^{1,2}, B. CABON¹

1. Airbus Defence and Space – Space Systems
31 rue des Cosmonautes – Z.I. du Palays
31402 Toulouse Cedex 4 - France
ludivine.boche-sauvan@astrium.eads.net,
bertrand.cabon@astrium.eads.net

MJ. HUGUET², E. HEBRARD³

2. Université de Toulouse – INSA - LAAS
31400 Toulouse - France
3. CNRS - LAAS - 7 avenue du colonel Roche
31400 Toulouse - France
huguet@laas.fr, hebrard@laas.fr

ABSTRACT: *The increasing complexity in telecommunication satellite payload design demands new test sequencing approaches to face the induced increase of validation complexity. In such a sequencing problem, each test requires specific payload equipment states at a dedicated temperature range. The industrial objectives are both to reduce the total time of tests and to keep the thermal stability of the payload. For solving this problem, we compare several strategies. The first one consists in packing compatible tests together and run them package after package. The second approach aims at masking the preparation time of tests and considers the sequencing of all tests separately. In this paper, we propose to improve the first approach by optimizing the sequence of compatible tests to reduce the total time. Then, we propose heuristic methods for the second approach. Experiments are conducted on real payload instances and show the respective interest of the two approaches.*

KEYWORDS: *Test sequencing problem, heuristics, telecommunication satellite.*

1 INTRODUCTION

Over the past decades, telecommunication demand has led to satellites with an increasing number of channels, and as a consequence the complexity of any task associated to the payload design, test or performance validation has increased. Before being delivered to the telecommunication operator and launched into orbit, the complete spacecraft must be tested to show its ability to fulfil its mission in the anticipated environment. The satellite must be tested under a simulated space environment under thermal vacuum created in a vacuum chamber. Two thermal phases are settled to test the payload performances: the COLD and the HOT phase. The HOT phase simulates the heat of the solar arrays on the satellite, and the COLD phase the moment when the satellite is in the Earth shadow. The satellite must perform under these two extreme environments. Performing a test requires the full payload, or a part of it, to be in a specific state and the temperature in the payload has to be stabilized. Then, for each test, the preparation times may be time consuming. This test phase is critical and expensive for the satellite manufacturer, who therefore aims at minimizing its duration.

The problem here is to find the best sequence of the tests in order to minimize the overall duration but also to keep a good thermal stability of the payload as it may lead to adding waiting time, delays. The payload is the whole set of equipment that will actually perform the mission of a satellite. In a telecommunication satellite, the payload gathers the different equipment which will be

crossed by an RF signal. The payload will receive, amplify and emit back the signal, but also route it between the different elements. The equipment is composed of:

- **active units**, which use external energy and dissipate heat while working such as Traveling Wave Tube Amplifier. The thermal stability of the payload only depends on changes on active units.
- **passive units among which switches** routing the signal among several ports connected to cables or wave guides.

The functional tests assess whether the payload equipment will be able to perform the mission or not. Such a test consists in entering an RF signal into the payload just behind the antenna to simulate a signal reception, and capturing back the signal just before the emitting antenna, once it went through the payload. A post analysis considering the entering and exiting signals will reveal whether the test was successful or not. If not, a deeper investigation may be set up to search which connection or equipment failed.

In order to model this, we define two steps for each test:

- a preparation step in which all active and passive units are put in a specific state;
- a running step where the test is executing on the payload. We consider that each running step has the same duration and that two tests cannot be run at the same time.

Before running, a test requires the RF path that will be used to be ready: the group of active units of this path must be turned ON (see below for instance in figure 1

rectangular units from 5005 to 5009), and the switches of this same path must be turned in dedicated position to route the signal through the right path (see in figure 1 round units in ranges of 3000s, 4000s and 7000s).

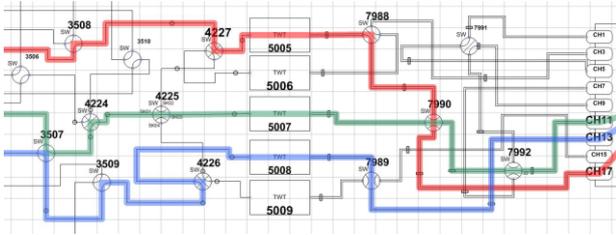


Figure 1: Small paths in a payload section

A preparation is therefore needed for each test, preparation that may induce delays when turning switches to the demanded position, but also when activating units which will need pre-heating or post-cooling times. Moreover, some thermal constraints limit the number of active units stated ON at the same time. In this study, we consider that preparation times for all passive units are identical and that preparation times for all active units are also identical, albeit significantly longer than for passive units.

Although there is only at most one test running at the same time, some tests can be in their preparation step when another test is running. For that reason, both running test and preparations tests have to respect constraints on switches (use of compatible switches avoiding switch position requirement conflict) and constraints on active units (maximum number of ON active units). Moreover, whenever a group of active unit is stated ON or OFF, it might result in thermal destabilization of the payload which might in turn entail important delays. Therefore, minimizing the number of changes in the status of active units helps ensuring the thermal stability of the system. The objective is then to sequence all the tests whilst minimizing both the total duration and the number of changes for the active units.

The remainder of this paper is organized as follows: In section 2, we present two main approaches for solving this test sequencing problem. In section 3, we propose a mathematical model, and in section 4 heuristic methods for these two approaches. In section 5 we give some experimental results on real instances, and their analysis in section 6.

2 SOLVING STRATEGIES

2.1 Packing strategy

This strategy consists of packing the tests into a minimal number of **configurations** to reduce the number of payload modifications where a configuration (or test family) is a full state of the payload defining any

switches position and any active equipment status (ON/OFF). We can distinguish two types of configurations:

- **active configurations** that define the status of the active equipment. Running consecutively two tests from a same active configuration will not need any active unit status change (but may need switches to be rotated).
- **passive configurations** that define the positions of the switches. Running consecutively two tests from a same passive configuration will not need any switch rotation (but may need active unit status to be modified).

Any test must be compatible with its allocated passive and active configuration, meaning allowing each switch to be in the position as defined by the configuration and every required active unit to be ON.

The packing strategy was studied by (Maillet, 2012) and considers two steps:

- packing the tests into active configurations. This first step will focus on thermal stability as it will minimize the number of active unit changing.
- packing the tests belonging to each active configuration into passive configurations. This will minimize the number of switches turning.

At each of these steps, an objective remains: to minimize the number of resulting configurations. This strategy is illustrated in figure 2, where red squares represent active configurations, green squares passive configurations and blue squares a test in its running step (during one time slot). Between two passive configurations, green time slots correspond to the preparation times for turning switches (one time slot in the example). Between two active configurations, the red time slots correspond to the preparation time for active units (three time slot in the example; that is longer than switches preparation times as active units demand pre-heating time). Note also that, during the preparation time for active units, switches can be turned. The two preparation times are then combined during one time slot in this figure 2 (time slot in red and green).

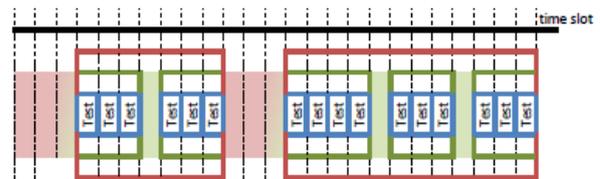


Figure 2: Example of a sequencing solution with the packing strategy

The packing strategy leads to minimizing the number of changes for both active and passive units. To this purpose, the number of configurations is minimized. However, between two configurations, many changes

may occur. This problem is not directly addressed in this approach.

A constraint-based model was proposed (Maillet, 2012) for the packing strategy using a tree search method with conflict directed backjumping, adaptive weighted variable heuristics and restart mechanisms. The objective function is to minimize the number of configurations.

2.2 Packing and sequencing strategy

Observe that if we run the tests grouped in configurations such as above in 2.1 (Maillet, 2012), changes on active and passive units are required when and only when going from a configuration to the next (active and passive, respectively). Therefore, the order in which configurations are run shall have no bearing on the overall duration. However, the total number of changes of status for active units does depend on this order, and this number of changes will actually influence thermal stability on the bench, leading to a further delay before reaching the required thermal condition of the next configuration. We therefore extend the approach introduced in (Maillet, 2012) in the following way: we build a graph with one vertex per configuration, and where the edge between two configurations is labeled with the number of changes of status for active units when going from configuration to the other. Then we solve a Traveling Salesman Problem (TSP) on this graph. The shortest tour indeed corresponds to the sequence of tests minimizing the number of changes.

2.3 Packing and sequencing bench

A bench of 100 tests has been run including random distribution noise in order to propagate differently from the search tree model from Maillet (2012). The noise changes the variables and values before launching the constraint satisfaction problem solver: the initial allocation of the paths and configurations are randomly changed before solving. The results will keep the same number of configurations but the paths will be differently allocated in these configurations. Then the time loss will still remain the same, however the total of active units changes will differ.

2.4 Unitary sequencing

As the duration of a test cannot be shortened, the time to be minimized is only between tests; that is, the preparation times or the reconfiguration times with the previous strategy. In the unitary sequencing strategy, we consider that each test constitutes its own configuration and we take advantage from the observation that some tests can be prepared while the current one is being run. In the best case, the sequencing solution, is such that all preparation times are fully masked by a running step and the number of time slots for running the tests is equal to

the number of tests. In figure 3, we represent a sequencing solution for a set of tests. The blue squares represent the running time of test; the green squares correspond to preparation times for switches and the red ones the preparation times for active units. As previously, switches can be turned during the preparation of active units (represented by red and green time slot).

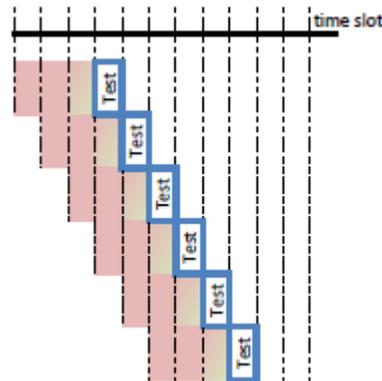


Figure 3: Example of a sequencing solution with the unitary sequencing strategy

In the unitary sequencing strategy, we want to minimize the number of time slots required for completing the sequence of tests and minimize the total number of times any active unit is switched ON or OFF.

In the next section, we will present a mathematical model for the test sequencing problem with the unitary sequencing strategy.

3 PROBLEM MODELLING

A payload is composed of a set \mathcal{A} of n_a active units and a set \mathcal{S} of n_s switches. One has to run a set \mathcal{P} of n_p RF test paths simply called tests in the following.

We define a set \mathcal{C} of time slots. We assume that all tests have the same duration. Then each time slot corresponds to the duration of a test. The number of time slots n_c is arbitrarily fixed (for instance twice the number of tests); in the best case it will be equal to the number of tests.

In each time slot c , only one test p can be run, but before running a test, active units and switches have to be put in the correct position: $k_{\mathcal{A}}$ denotes the number of time slots for the preparation of active units and $k_{\mathcal{S}}$ is the number of time slots for the preparation of switches. We assume that preparation times are the same for all active units and all switches respectively and that these numbers of time slots do not vary with the number of active units (respectively switches) involved in the preparation.

If an active unit is required for a test, then it shall be in state ON for running this test. If a switch is on a test, then it shall be in a specific position, as defined by the

test (in practice, there are several types of switches, each one having a given number of position). Then, for each test p run in the time slot c , $Req_c(a)$ tells if the active unit a is required in c and $Sw_p(s)$ gives the position of the switch s as required by p .

To model the test sequencing problem with the unitary sequencing strategy, we introduce several decision variables:

- an integer vector X , $\forall p \in \mathcal{P}$, such that $X_p = c$ if the time slot c is associated to the test p .
- a Boolean matrix Y , $\forall a \in \mathcal{A}, \forall c \in \mathcal{C} : Y_{ac} = 1$ if the active unit a is set ON in the time slot c , and $Y_{ac} = 0$ if the active unit a is set OFF in the time slot c .

The constraints are the following:

- **Allocation constraints:** A test must be allocated in one and only one time slot.

$$\forall p \in \mathcal{P}, \exists! c \in \mathcal{C} : X_p = c$$
- **Thermal constraints:** A thermal constraint gives for a subset $\hat{\mathcal{C}}$ of time slots and for a subset $\hat{\mathcal{A}}$ of active units an upper bound UB which shall bound the number of these active units which are ON.

$$\forall c \in \hat{\mathcal{C}}, \sum_{a \in \hat{\mathcal{A}}} Y_{ac} \leq UB.$$

- **Switch compatibility constraints:** As a switch may have only one position per time slot, two tests demanding different positions for the same switch may not impact a common time slot (to be run or prepared in).

For two tests p_1 and p_2 , if it exists a switch s such that $Sw_{p_1}(s) \neq Sw_{p_2}(s)$ then

$$X_{p_2} \notin \llbracket X_{p_1} - k_s, X_{p_1} + k_s \rrbracket.$$

Our objective is to minimize both the number of ON/OFF modifications for active equipment to obtain a better thermal stability of the payload and to minimize the total duration of tests, which corresponds to the number of time slots. The minimization of the number of changes is represented by:

$$\sum_{c=0}^{n_c-1} \left(\sum_{a \in \mathcal{A}} |Y_{a,c+1} - Y_{a,c}| \right)$$

For the minimization of the total duration, we aim at minimizing the number of used time slots n_c .

4 HEURISTIC METHODS

4.1 General principles

In this section, we present a simulated annealing method that aims to solve the test sequencing problem with the unitary sequencing strategy.

The simulated annealing is a heuristic based on probabilistic search, derived from metallic crystals behavior when controlling its temperature (Kirkpatrick et al,

1983). It starts with an initial solution and visits its neighborhood. At the beginning of the method, the probability of accepting a worse neighbor is high and during the method, due to the decreasing of temperature, this probability decreases leading to a classical hill climbing method. The method stops when given conditions are met, for instance a time out condition. The pseudo code of simulated annealing is given in Algorithm 1.

In our method, the temperature decreases step by step by a factor K (with $K < 1$), the value of K is adjusted for each instance thanks to its characteristics and for each given time out. The initial temperature T_0 is obtained by a preprocessing phase of the simulated annealing that finds a temperature such that sufficient number of neighbors is accepted.

```

currentSol ← initialSol
bestSol ← initialSol

K      (decreasing factor)
T ← T0 (initial temperature)
while not stoppingCondition() do
  neighborSol ← generateNeighbor()
  if acceptNeighbor() then
    currentSol ← neighborSol
    if Criteria(neighborSol) <
       Criteria(bestSol) then
      bestSol ← currentSol
  if decreasingCondition() then
    T ← T * K
end while
return bestSol

```

Algorithm 1. Simulated Annealing

In this algorithm:

- The function `decreasingCondition()` corresponds to a fixed number of iterations per step.
- The function `acceptNeighbor()` is the classical Metropolis condition used in simulated annealing:
`acceptNeighbor` is true if
 $CriteriaNeighbor < CriteriaSolution$
OR
 $\lambda < e^{\frac{(CriteriaCurrent - CriteriaNeighbor)}{T}}$

With λ a random value in $[0,1]$.

4.2 Initial solution

The initial solution is directly built from the actual tests list as given by the validation team (the team that processes the tests): $initialSol = \{1, 2, 3, \dots, n_p\}$.

In practice, the initial solution based on the industrial experience leads to a correct evaluation (few numbers of incompatibilities, constraint violations and changes). In order to avoid a local optimum, this solution is randomly moved via iterative swaps (iterative exchange of 2 tests in the sequence).

4.3 Temperature initialization

From a temperature of $T=0.1$, a simulated annealing loop is used to heat instead of cooling. The temperature is increased by 1.5 at each step of iterations, until the metropolis condition accepts 90% of the neighbors. The hottest temperature reached is the initial temperature T_0 to start the cooling phase of the heuristic.

4.4 Cooling phase and stopping condition

The final temperature, according to the Metropolis condition, will settled the last probability of accepting degrading neighbors. In order to end with a convergent solution, this final temperature is set to $T_f = 10^{-12}$.

The heating phase as described in section 4.3 will able to find the speed of the algorithm: its ability to run a certain number of Metropolis steps over duration, for this particular instance.

From this *speed* the initial and final temperatures, and a time limit condition that we fix, the decreasing coefficient of the temperature may be calculated (the temperature observe a decreasing geometric suite with this coefficient) such that the simulated annealing will perform the cooling between the two temperatures under the definite duration.

$$\alpha_{temperature\ decreasing} = \left(\frac{T_f}{T_0}\right)^{speed \cdot duration}$$

Our stopping condition will therefore be the duration of the cooling phase.

4.5 Heuristic global criterion

As the number of incompatibilities for switches and the number thermal constraint violations impact the length of the sequence, we use these values to evaluate a solution in the heuristics. The number of changes in active units will influence the thermal stability. Then our criterion is composed of three parts:

- the compatibility criterion C_{cpt} that counts the number of switch position conflicts which will be faced at a same time slot between preparing or running tests according to the resulting solution sequence;
- the thermal criterion C_{thl} that represents the number of active units in position ON over the different thermal constraints limitations;
- the number of changes in active units criterion C_{act} that counts the number of changes among active units between a time slot and the next one.

The global criterion results from a weighted sum of these three criteria:

$$C_{global} = \alpha C_{cpt} + \beta C_{thl} + \gamma C_{act}$$

where $\{\alpha, \beta, \gamma\}$ is the set of weights that will be defined for the experiments. This criterion is the one used for accepting or not a generated neighbor via the Metropolis condition.

The total duration for a sequence of tests is then evaluated to the total number of tests n_p plus the time needed to satisfy the compatibility constraints, $k_s C_{cpt}$ plus the time needed for satisfying the thermal constraints, $k_A C_{thl}$.

4.6 Neighborhoods

We propose two kinds of neighborhoods for this test sequencing problem. The first one consists in swapping tests in the sequence and the second one concerns the active units and aims to change the status of an equipment (from ON to OFF or the contrary) if it is possible (mandatory active units for a test cannot be put OFF).

In our problem, the status of each active unit is defined through:

- test requirement: $Req_c(a) = 1$ if the active unit a is mandatory at slot c ;
- stability requirement: $Set_c(a) = 1$ if the active unit a is set ON at slot c ;

Then, $Status_c(a) = Req_c(a) \vee Set_c(a)$: the unit is ON when used by the running or a preparing test, but may also be kept ON to avoid useless thermal disturbance.

For the first neighborhood, called **swap**, we consider only a swap among two random tests in the sequence. For the second neighborhood, called **change**, we consider:

- **change1**: that chooses randomly an active unit a , a slot c and a length l and changes values $Set_x(a)$ for all slots x from time slot $c + 1$ to time slot $c + l$ by the value $Req_c(a)$.
- **change2**: that chooses randomly an active unit a , a slot c and considers the value $Req_c(a)$.
 - o If $Req_c(a) = 1$: Set c' the last slot in a row after c for which $Req_{c'}(a) = 1$ and c'' the next one with $Req_{c''}(a) = 1$. Between c' and c'' no test requires a , change2 forces to keep a activated: $Set_x(a) = 1$ for any x test between c' and c'' .
 - o If $Req_c(a) = 0$: Set c' the first previous slot before c for which $Req_{c'}(a) = 1$ and c'' the next one with $Req_{c''}(a) = 1$. Between c' and c'' no test requires a , change2 relax the activation of a : $Set_x(a) = 0$ for any x test between c' and c'' .

In the simulated annealing heuristic, we fix a ratio between swap and change neighborhoods, for instance with a ratio equal to 20%, we use 20% of changes and 80% of swaps.

5 EXPERIMENTS

5.1 Objectives

The goal in the experiments is the comparison of the different approaches over the two objectives:

- minimize the global duration of the sequence
- minimize the number of active units' activation and deactivation to avoid further waiting time for global payload thermal stabilization.

5.2 Experimental context

An experimental study had been performed with 3 different satellite architectures, in cold and hot environment leading to 6 realistic test cases.

- Satellite A: 85 switches, 40 active units and 300 tests.
- Satellite B: 450 switches, 130 active units and 400 tests.
- Satellite C: 370 switches, 100 active units and 800 tests.

All these satellites are tested under two thermal phases: HOT and COLD. In each of these two phases the same list of tests has to be carried out, but the thermal constraints will be different.

The simulated annealing was written in Java and the experiments have been run on an Intel® Core™ i5-4300U CPU 1.90GHz 2.50GHz, 8.00Go RAM under Windows 7.

We apply the simulated annealing with with 30 minutes time out stopping condition, three different ratios 20%, 50% and 80% among swaps and changes, 10000 iterations per step and the two combinations of neighborhoods (swap-change1 or SC1; swap-change2 or SC2) with 30 minutes time out stopping condition.

Three different weighted systems $\{\alpha, \beta, \gamma\}$ have been used to solve the complete problem:

- W1 = {1,10,5}: Focused on thermal side: the thermal constraint shall be fulfilled to avoid time laps in waiting for a unit to be at the demanded temperature, and focused on thermal stability.
- W2 = {10,10,5}: Focused on masking preparing times. If the compatibility and thermal constraints are fulfilled then no time is wasted, the tests may always be running.
- W3 = {0,10,5}: Focused minimizing on active units changes under the thermal constraints.

5.3 Experimental evaluation

In this subsection, the calculation of the different objectives for each of the strategies is described.

5.3.1 Total duration and time loss

The total duration of the sequence is given in number of time slots. To compare the two strategies, we evaluate for each of them the number of time slots needed for preparation time and not for running a test. We call the unmasked preparation time the "time loss". In our experiments, the time loss is given as the percentage of the unmasked preparation (n_{loss}) over the full sequence duration, i.e. the testing (n_p) plus the unmasked preparation slots.

$$T_{loss}(\%) = \frac{n_{loss}}{n_{loss} + n_p} * 100$$

In packing strategy, the time loss might be simplified as the summation of the time needed between each active configuration added to for each of these active configurations, the time needed between all the passive configurations.

Denoting n_{c_a} the number of active configurations and n_{c_a, c_p} the number of passive configurations in the configuration c_a , and n_p and the number of tests, the time loss is calculated with the following equation:

$$n_{loss \text{ for packing}} = k_A * (n_{c_a} - 1) + \sum_{c_a \in C_a} k_S * (n_{c_a, c_p} - 1)$$

In this strategy, the changes from a configuration to another are assumed equal, therefore their order is not important and the time loss in packing and sequencing strategy is the same.

In practice, the preparation time needed between two active or passive configurations may depend to the number of changes. We choose to consider that preparation times are identical whatever the number of involved units to have a lower bound of the real expected duration.

In the unitary sequence strategy, the time loss is considered as the weighted summation of the different compatibility and thermal conflicts.

$$n_{loss \text{ for unitary}} = k_A * C_{thl} + k_S * C_{cpt}$$

Note that this calculation is an upper bound of the exact total duration as: several conflicts may be solved during the same empty slots.

In our experiments, the time slot needed to turn switches to prepare a test, masked or unmasked, is assumed as $k_S = 1$ slot. The time needed to activate and pre-heat an active unit is assumed as $k_A = 5$ slots.

5.3.2 Thermal stability: number of active unit changes

In the packing and packing+sequencing strategies, the active unit changes are given between the different active configurations.

This criterion is the only one optimized between the packing and packing + sequencing strategies. Given the short number of active configurations, the best sequence regarding the minimization of active changes has been found after comparing all the possible configurations sequences via iterative configurations permutations.

5.3.3 Thermal stability: changing ratio

In addition to the considered criteria, we consider another evaluation of the thermal stability in our results. The thermal stability is the ability of minimizing the thermal gradients in the payload. We consider that the thermal gradient corresponds to the number of changes occurring in the same time, and simplify it as the “changing ratio”: this is the ratio between the total number of changes over the number of moments when at least one active unit change happens. In the case of packing or packing + sequencing strategies, this number of changing moments is actually the number of active reconfigurations.

5.4 Results

The detailed results from the simulated annealing are given in the appendixes: these are the final values of the criteria under the different weights and three different neighboring ratios 20%, 50% and 80% (this ratio is explained section 4.6).

These detailed results give the following observation: the first neighboring, less directed, is finally more adequate to be used with the simulated annealing heuristic. SC1 is then the one used in the following table and analysis, and the neighboring ratio is fixed to 50%.

We compare then the following methods: the first one is the packing method, the second one is the packing + sequencing method, the last ones are the simulated annealing with the change 1 (SC1) and ratio 50%, and with the three weighting systems W1, W2 and W3. The results are given in Table 1. Each row corresponds to a given satellite in a thermal phase (cold or hot).

For each method, we give the values of the time loss, the number of active changes and the changing ratio (as described in section 5.3.3). Bold values indicate the best found values.

	Packing			Packing+seq			Unitary W1, 50%			Unitary W2, 50%			Unitary W3, 50%		
	Time loss (%)	Active changes	Changing ratio	Time loss (%)	Active changes	Changing ratio	Time loss (%)	Active changes	Changing ratio	Time loss (%)	Active changes	Changing ratio	Time loss (%)	Active changes	Changing ratio
Sat A Cold	13,14	136	27,2	13,14	62	12,4	10,00	64	1,16	3,57	103	1,27	57,70	57	1,21
Sat A Hot	8,73	50	16,7	8,73	48	16	7,43	52	1,18	4,14	54	1,13	58,57	70	1,21
Sat B Cold	19,62	410	45,6	19,62	362	40,2	0	263	1,41	0	276	1,5	30,31	270	1,51
Sat B Hot	11,08	242	60,5	11,08	228	57	0	208	1,34	0	225	1,4	25,64	218	1,38
Sat C Cold	14,14	299	42,7	14,14	251	35,9	0	272	1,19	0	281	1,28	28,15	273	1,29
Sat C Hot	6,94	92	46	6,94	89	44,5	0	235	1,21	0	217	1,14	28,15	228	1,12

Table 1: Experiments results

6 RESULTS ANALYSIS

6.1 Time loss

The packing strategy, even well sequenced, keeps significant time loss due to configurations preparations. Furthermore, the calculated time loss stays under the operational one as changing the status of a full payload induces high thermal disturbances, forcing to wait for its stability.

A reverse engineering study on packing strategy in operations (Boche-Sauvan, 2012) has shown that considering reconfiguration and testing time, about 30% was taken by unmasked reconfiguration time for 70% in running tests. The Unitary W3 is here the only one above such a result: this is due to the many switches conflicts as the passive criterion is not considered. In all the other models, taking in account all the different sides of the problem, the time loss is even under 20% of the total sequence time.

The best values for time loss are obtained with the unitary sequence approach with weights W1 and W2 (as it concerns all the sides of the problem): most of the cases cope with masking the preparation times. A few added slots remain for the satellite A. From the annexes we extract all the results with change 1 for satellite A and compare the different values of the time loss according to the neighboring ratio (see the following table).

Satellite A		Cold	Hot
20%	W1	7,4%	6,9%
	W2	3,6%	0,0%
50%	W1	10,0%	7,4%
	W2	3,6%	4,1%
80%	W1	14,7%	5,8%
	W2	4,1%	5,3%

Table 2: Time loss with Sat A under unitary sequence

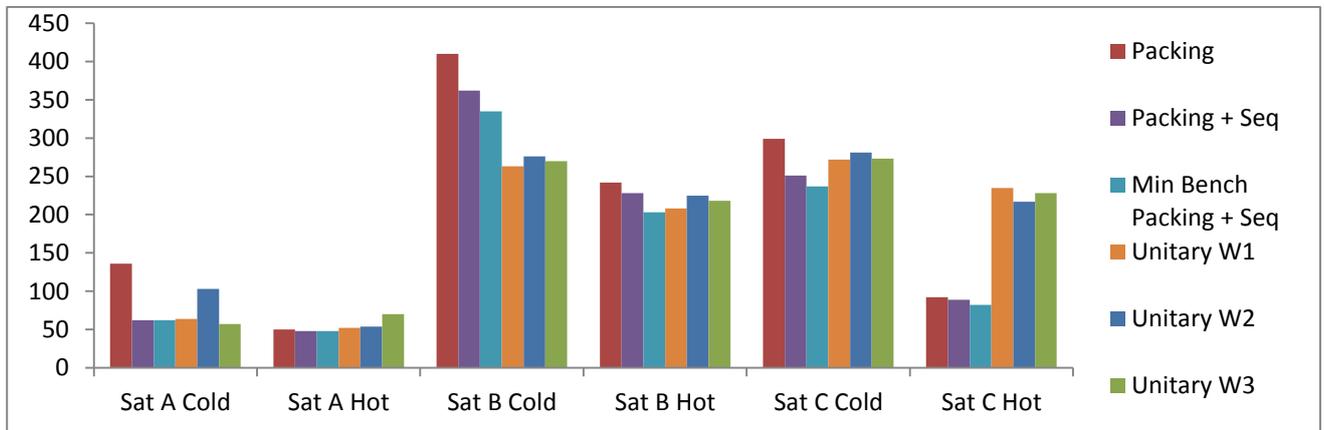


Figure 4: Active changes comparison over the approaches

In figure 4, the “Min Bench Packing + Seq” is the minimal value found via the bench of 100 iterations (see section 5.5).

The unitary sequencing strategy does not minimize active changes as much as the packing approach, even if it is part of its global criterion. The best approach in terms of global number of active unit status changes is still the packing one, as while packing the tests, it packs the changes too.

6.3 Thermal stability

Minimizing the number of active units changes does not necessarily mean gaining in minimizing the changing ratio (that improves the thermal stability) as the results shows us, see for example in Table 1 the results in satellite C: the best approach for active changes is packing + sequencing whereas the best one for thermal stability is the unitary sequence. Even for the other case, the unitary

In any case in this table, the preparation time with the unitary sequencing strategy is better than the one experienced in operation, which is compliant with its goal in masking this time.

6.2 Number of active units changes

The best values for the number of changes depend on the considered satellite. For satellites A and C, the packing+sequencing strategy is the best one, for satellite B, the unitary strategy W1 obtains the best results. We remark that only one value of active units changes is the best with the unitary sequence W3, and not far from the W1 result. The unitary sequence shows certain robustness on the compatibility impact considering the number of active unit changes.

approach demands far less changes at the same time than the other approaches.

Let us analyze the worst case for the unitary sequencing strategy in section 6.2, the satellite C in Hot phase. The simulated annealing W1 result is far from any of the other approaches results in term of active unit changes. However it will avoid any high thermal disturbance as it demands local and small active changes all along the sequence. To illustrate the thermal stability, Figure 5 is a graph of the changes per slot with unitary approach, and Figure 6 the one with packing approach.

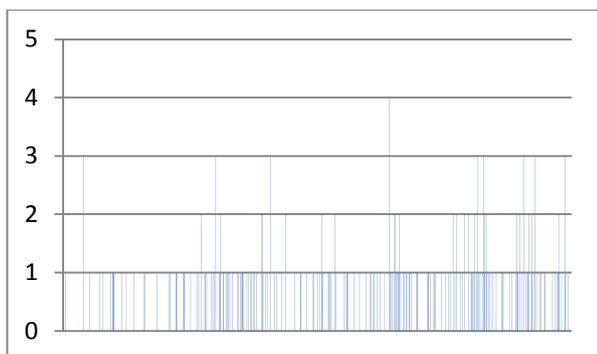


Figure 5: Number of changes per slot with unitary sequencing approach, case sat C Hot.

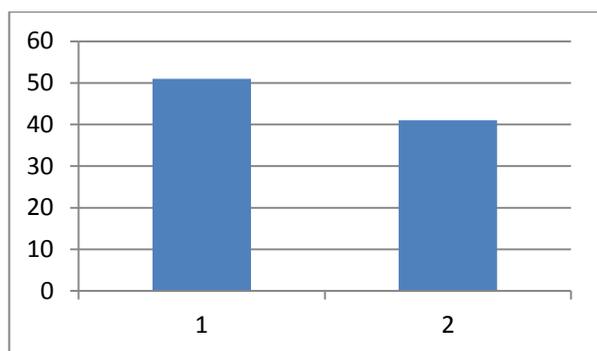


Figure 6: Number of change per reconfiguration with the packing strategy, case Sat C Hot

Satellite C has 100 active units and requires only 3 active configurations under hot phase, then 2 reconfigurations. Obviously, the packing strategy induces high thermal evolutions as it changes from 40% to 50% of the payload active units between the configurations, whereas the unitary sequencing approach shall help in mastering thermal stability through small and local changes, no more than 4% of the active units.

Obviously, the packing strategy induces high thermal evolutions, whereas the unitary sequencing approach shall help in mastering thermal stability.

7 CONCLUSION

In this paper, a mathematical model, a new approach and a heuristic have been proposed to solve a test sequencing problem. This new approach, unitary test sequencing, does not necessarily succeeds in minimizing active units changes, however it succeeds in the main aims: minimizing the global duration and enhancing thermal stability.

Moreover, the new unitary test sequencing enhances the thermal stability. This strategy requires frequent small active units status modifications; this shall challenge the operational process on the test bench. The first approach, packing the test, has the advantage of creating, once the payload is thermally stable, long and stable sequences with no thermal questioning. Nevertheless, the following improvement on operation tools, targeting to master the dynamics of unitary sequence strategy, will become a real asset for the next telecommunication payload satellites.

REFERENCES

- Boche-Sauvan L. 2012, *Telecom satellites validation process evaluation*, Internship ISAE-Astrium, Toulouse, France.
- Kirkpatrick S., Gelatt C. D. Jr., Vecchi M. P., 1983, *Optimization by Simulated Annealing*, *Science*, Vol.220, Number 4598.
- Maillet C. 2012, *Optimisation des plans de test des charges utiles des satellites de télécommunication*, PhD Onera - Astrium, Toulouse, France.

APPENDIXES

Change 1		20%			50%			80%		
		w1	w2	w3	w1	w2	w3	w1	w2	w3
A Cold	Ccpt	26	12	486	36	12	442	56	14	462
	Cthl	0	0	0	0	0	0	0	0	0
	Cact	76	62	66	64	103	57	74	91	76
A Hot	Ccpt	24	0	458	26	14	458	20	18	450
	Cthl	0	0	0	0	0	0	0	0	0
	Cact	48	63	62	52	54	70	79	73	51
B Cold	Ccpt	0	0	150	0	0	164	0	0	148
	Cthl	0	0	0	0	0	0	0	0	0
	Cact	290	270	267	263	276	270	279	296	256
B Hot	Ccpt	0	0	136	0	0	130	0	0	138
	Cthl	0	0	0	0	0	0	0	0	0
	Cact	216	212	228	208	225	218	219	215	214
C Cold	Ccpt	0	0	298	0	0	326	0	0	326
	Cthl	0	0	0	0	0	0	0	0	0
	Cact	281	294	273	272	281	273	303	279	276
C Hot	Ccpt	0	0	346	0	0	326	0	0	364
	Cthl	0	0	0	0	0	0	0	0	0
	Cact	257	260	228	235	217	228	263	274	214

Table 3: Simulated annealing results with Change 1

Change 2		20%		50%		80%	
		w1	w2	w1	w2	w1	w2
A Cold	Ccpt	92	6	96	14	112	8
	Cthl	0	0	0	0	0	1
	Cact	140	194	142	188	142	204
A Hot	Ccpt	80	8	110	18	92	20
	Cthl	0	1	0	0	0	0
	Cact	142	202	135	178	135	193
B Cold	Ccpt	0	0	0	0	8	0
	Cthl	0	0	0	0	0	0
	Cact	405	404	416	413	407	436
B Hot	Ccpt	6	0	2	0	4	0
	Cthl	0	0	0	0	0	0
	Cact	364	414	366	437	414	444
C Cold	Ccpt	4	0	2	0	10	0
	Cthl	0	0	0	0	0	0
	Cact	619	626	653	643	731	714
C Hot	Ccpt	0	0	2	0	12	0
	Cthl	0	0	0	0	0	0
	Cact	584	612	581	622	695	737

Table 4: Simulated annealing results with Change 2