New Results for the GEO-CAPE Observation Scheduling Problem

Philippe Laborie and **Bilal Messaoudi**
IBM Analytics
IBM France Lab, Gentilly, France

Abstract

A challenging Earth-observing satellite scheduling problem was recently studied in (Frank, Do and Tran 2016) for which the best resolution approach so far on the proposed benchmark is a time-indexed Mixed Integer Linear Program (MILP) formulation. This MILP formulation produces feasible solutions but is not able to prove optimality or to provide tight optimality gaps, making it difficult to assess the quality of existing solutions. In this paper, we first introduce an alternative disjunctive MILP formulation that manages to close more than half of the instances of the benchmark. This MILP formulation is then relaxed to provide good bounds on optimal values for the unsolved instances. We then propose a CP Optimizer model that consistently outperforms the original time-indexed MILP formulation, reducing the optimality gap by more than 4 times. This Constraint Programming (CP) formulation is very concise: we give its complete OPL implementation in the paper. Some improvements of this CP model are reported resulting in an approach that produces optimal or near-optimal solutions (optimality gap smaller than 1%) for about 80% of the instances. Unlike the MILP formulations, it is able to quickly produce good quality schedules and it is expected to be flexible enough to handle the changing requirements of the application.

1 Introduction

A challenging Earth-observing satellite scheduling problem was recently proposed in (Frank, Do, and Tran 2016). The problem consists of scheduling observations of a set of different scenes which are parts of oceans bordering the United States. Those observations are processed by a single dedicated imaging instrument present on the satellite in order to perform ocean color remote sensing. To achieve this goal, observations of a given scene must occur multiple times during the schedule horizon. Different classes of scientific objectives are considered where ideal temporal separations between consecutive observations of a given scene are defined depending on the type of instrument used. Times during which scenes can be observed are constrained by the available daylight and by the cloud coverage which changes throughout the day. In addition, the instrument can scan only one scene at a time. Scheduling consists of choosing which scene to be observed at each time of the day, maximizing the gains that result from temporal separations lengths, while satisfying all the constraints.

In (Frank, Do, and Tran 2016), the authors proposed a Mixed Integer Linear Program (MILP) and a Constraint Programming (CP) model to tackle this problem. The two formulations were applied on realistic instances (with real cloud coverage data). The reported results show that the MILP model performs better than the CP model in terms of total gain (objective function). Both MILP and CP models were solved using CPLEX and CP Optimizer respectively within a one hour time limit. The authors noticed that the addressed problems are too large and complex to be solved to optimality. Another analysis was performed to compare the quality of the two instruments (FR and COEDI) which showed that schedules for FR instrument have in general better quality than those for COEDI.

In the present paper, we first introduce some concepts and properties of the problem that will be exploited by our models. We then present a MILP disjunctive model that is able to solve more than half of the instances of the GEO-CAPE scheduling benchmark to optimality. Unfortunately, this model produces poor quality solutions for the problems that cannot be solved to optimality. We show how the model can be relaxed to produce upper bounds for the remaining challenging problems. These upper bounds are used to evaluate and compare the quality of the different solutions. Next, we describe a new CP model based on the scheduling concepts of CP Optimizer (Laborie 2009). This model is very concise: we give its complete OPL formulation in the paper. Experiments using the automatic search of IBM ILOG CP Optimizer 12.7 show that this model consistently outperforms the original MILP formulation on the most challenging instances (optimality gap is divided by a factor greater than 4) while it produces near-optimal solutions for the other instances. Building up this basic CP formulation, we present some improvements that result in a reduction of the optimality gap by a factor greater than 6 compared to the original MILP and an approach that produces optimal or near-optimal solutions (optimality gap smaller than 1%) for about 80% of the instances. Beside pure performance, another advantage of the CP models is that they can be easily adapted to handle the changing requirements of the actual application.

2 The GEO-CAPE Scheduling Problem

In this section, we recap the GEO-CAPE observation scheduling problem originally described in (Frank, Do, and Tran 2016). We are given a set Ψ of scenes to be observed and a set of timeslots H representing the schedule horizon. Each observation of a given scene takes one timeslot. Each scene $i \in \Psi$ is characterized by a baseline (S_B^i) and a threshold separation value (S_T^i), together with the schedule quality values V_B^i and V_T^i when the separation between two consecutive observations of scene i is equal to S_B^i and S_T^i respectively. This separation value must be always greater or equal to the minimum baseline separation. More formally, the value $V(d)$ of two consecutive observations of the same scene i separated by a duration d such that $S_B^i \leq d \leq S_T^i$ is defined as:

$$V(d) = \alpha_i \cdot d + \beta_i \tag{1}$$

Where:

$$\alpha_i = \frac{V_T^i - V_B^i}{S_T^i - S_B^i} \text{ and } \beta_i = V_B^i - \left(S_B^i \cdot \frac{V_T^i - V_B^i}{S_T^i - S_B^i} \right)$$

The value $V(d)$ is zero when $d > S_T^i$. Function $V(d)$ is illustrated in Figure 1.

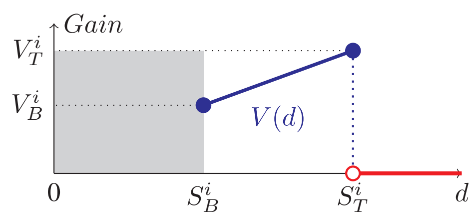

Figure 1: Objective value gain $V(d)$ between two consecutive observations as a function of separation time d

The objective considered in the GEO-CAPE scheduling problem is to maximize the total gain generated from consecutive observations. Note that the first observation of a scene does not provide any gain by itself.

Scenes are not always observable. The available daylight and the cloud coverage of a scene $i \in \Psi$ are translated into a set of observable timeslots $C^i \subset H$. Each observation of scene i must be scheduled at a timeslot belonging to C^i.

Finally, the instrument can observe only one scene at a time so all observations must be scheduled at different timeslot.

An example solution for an instance of the GEO-CAPE scheduling problem is shown in Figure 8 where each scene is represented by an horizontal line. The scheduled observations appear as solid squares whereas empty squares represent the possible timeslots. In this instance $[S_B^i, S_T^i] = [8, 16]$ and $[V_B^i, V_T^i] = [0.6, 1.0]$, so for example the separation time between the first two observations of the first scene on the top is 16 (equal to S_T^i) so it contributes with a value $V_T^i = 1.0$ to the total gain. The very last line represents the union of all scheduled (non-overlapping) observations.

3 Problem Properties

In a solution, for a given scene with n observations, let a_j denote the timeslot of the j^{th} observation ($1 \leq j \leq n$). For $1 \leq s \leq e \leq n$, we denote $a_{[s,e]} = [a_s, a_{s+1}, ..., a_e]$ the ordered set of successive observations between a_s and a_e.

Informally speaking, we say that a set of successive observations $a_{[s,e]}$ is a *train* of observations if all separation times between successive observations in the set contribute with a strictly positive value. We say that $a_{[s,e]}$ is a *block* of observations if it is a maximal *train* in the sense of inclusion. A block of cardinality 1 is called an *isolated* observation.

Definition 1 (Train of observations) *Given a schedule of consecutive observations $(a_j)_{j\in[1,n]}$ for a given scene, a subset of consecutive observations $a_{[s,e]}$ with $1 \leq s \leq e \leq n$ is said to be a* train *if and only if:*

$$(s = e) \vee (\forall j \in [s+1, e], S_B \leq a_j - a_{j-1} \leq S_T)$$

Definition 2 (Block of observations) *Given a schedule of consecutive observations $(a_j)_{j\in[1,n]}$ for a given scene, a train of observations $a_{[s,e]}$ with $1 \leq s \leq e \leq n$ is said to be a* block *if and only if it satisfies the 2 following conditions:*

$$(s = 1) \vee (a_s - a_{s-1} > S_T)$$

$$(e = n) \vee (a_{e+1} - a_e > S_T)$$

Definition 3 (Isolated observation) *An isolated observation is a block of observations of cardinality 1.*

Isolated observations and blocks are illustrated in Figure 2 for a scene with $S_T = 5$. Observations are depicted with squares, the number under each observation represents its date. In this solution, a_5 is an isolated observation as its separation times with the previous observation (19-13=6) and with the next observation (26-19=7) are both greater than $S_T = 5$. Besides the isolated observation $a_{[5]}$, this solution has 2 observation blocks of cardinality greater than 1: $a_{[1,4]}$ and $a_{[6,7]}$

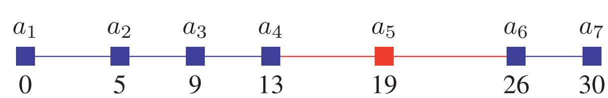

Figure 2: Isolated observations and blocks in a solution for a given scene

Proposition 1 (Isolated observations uselessness) *Let S be a feasible solution of the GEO-CAPE scheduling problem. Removing all isolated observations from S results in a feasible solution with the same objective value.*

The proof is trivial. A corollary of this proposition is that a constraint forbidding the presence of isolated observations can be added as a dominance rule in the model (Jouglet and Carlier 2011). In the sequel of the paper we suppose all trains and blocks of observations have a cardinality greater than 1.

Proposition 2 (Value of a train of observations) *Let $a_{[s,e]}$ with $s \leq e$ be a train of observations for scene i. The total* *value of the train only depends on its duration $d = a_e - a_s$ and its cardinality $c = e - s + 1$:*

$$V(a_{[s,e]}) = \alpha_i \cdot d + \beta_i \cdot (c-1)$$

The proof is a direct consequence of the linearity of the formula for V(d) in Equation 1. In particular, the proposition shows that the total value of the train does not depend on the position of the observations a_j with $s < j < e$ inside the train.

For a scene i, we can notice that if there exist some non-observability time window of length greater than S_T^i, then the scene can be decomposed into two different sub-scenes without impacting the solutions of the problem. This corresponds to the notion of *pseudo-scene* formalized below. Furthermore, any scene i such that $\max(C^i) - \min(C^i) < S_B^i$ can clearly be ignored as it does not provide enough separation time to accommodate more than an isolated observation.

If C is a non-empty ordered set of integers, we denote: $|C|$ its cardinality and for all k in $[1, |C|]$, $C[k]$ denotes the k^{th} element of C.

Definition 4 (Pseudo-scene) *Let C^i be the ordered set of timeslots of scene i and $\Pi^i = (C_1^i, ..., C_p^i)$ the unique partition of C^i such that:*

$$\forall j \in [1,p], \forall k \in [1, |C_j^i| - 1], C_j^i[k+1] - C_j^i[k] \leq S_T^i$$

$$\forall j \in [1, p-1], C_{j+1}^i[1] - C_j^i[|C_j^i|] > S_T^i$$

Each element C_j^i of the partition Π^i such that $\max(C_j^i) - \min(C_j^i) \geq S_B^i$ defines a pseudo-scene (i,j) where the set of timeslots C^i of the original scene i is replaced by C_j^i.

Proposition 3 (Decomposition in pseudo-scenes) *The original problem is equivalent to the problem where each scene i is replaced by the set of pseudo-scenes (i,j) derived from scene i.*

The proof relies on the fact that a gap of length greater than S_T^i between consecutive timeslots of a given scene i cannot be overlapped by any train of observations: any feasible solution to the original problem is a feasible solution to the decomposed problem and vice-versa. Furthermore as separation times greater than S_T^i do not bring any gain, the decomposition does not impact the objective value.

As an illustrative example, if we consider the third last scene depicted in Figure 8, looking at the actual data we would see that the original timeslots for this scene are the four observability intervals $\{[13,24], [33,88], [122,185], [210,218]\}$. For this scene we have $[S_B^i, S_T^i] = [8,16]$. The non-observability gaps between possible observation intervals $[33,88]$ and $[122,185]$ (gap=33) and between possible observation intervals $[122,185]$ and $[210,218]$ (gap=24) are both larger than S_T^i thus the scene would be decomposed into three pseudo-scenes with respective observability windows: $\{[13,24],[33,88]\}$, $\{[122,185]\}$ and $\{[210,218]\}$. None of these pseudo-scenes can be eliminated because they all span an interval larger than $S_B^i = 8$. In the depicted solution, no observations are scheduled for this scene.

4 Optimal solutions and Upper bounds

In the original paper, the authors mention the possibility of modeling the problem as a disjunctive MILP (Ku and Beck 2016). We present such a model in the first part of this section. As we will see, this MILP turns out to be efficient enough to solve the easiest instances to optimality (mainly, the ones related with the FR instrument) but it provides poor quality solutions for the other problems. In the second part of the section we show how this model can be exploited to compute upper bounds for the unsolved instances.

4.1 A disjunctive MILP model

The model exploits proposition 3: the original scenes of the problem are decomposed into a set of pseudo-scenes. By abuse of notation, we denote Ψ the set of all the pseudo-scenes and, for a pseudo-scene $i \in \Psi$, C^i denotes the timeslots within the schedule horizon H where pseudo-scene i is observable.

The set of non-observable timeslots of a pseudo-scene i is the set $\bar{C}^i$ defined as:

$$\bar{C}^i = \left[\min(C^i), \max(C^i)\right] \setminus C^i$$

This set can be expressed as a set of maximal contiguous intervals of non-observability NoObs^i.

A_i denotes the set of possible observations of pseudo-scene i. As in the original MILP, the bound we use for the number of observations is :

$$|A_i| = \left\lfloor \frac{\max(C^i) - \min(C^i)}{S_B^i} \right\rfloor$$

A_i^* denotes the set of all possible observations of pseudo-scene i except the last one.

The following decision variables are used in the disjunctive MILP formulation:

- $o_{i,j}$: A binary variable indicating that the j^{th} observation of pseudo-scene i is processed.
- $b_{i_1,j_1;i_2,j_2}$: A binary variable indicating that observation j_1 of pseudo-scene i_1 occurs before observation j_2 of pseudo-scene i_2.
- $b_{i,j;k}$: A binary variable indicating that observation j of pseudo-scene i occurs before[1] the k^{th} interval of non-observability in NoObs^i.
- $t_{i,j}$: An integer variable indicating the timeslot value of the j^{th} observation of pseudo-scene i.
- $d_{i,j}$: An integer variable indicating the separation time between the j^{th} and the $j+1^{\text{th}}$ observation of pseudo-scene i.
- $z_{i,j}$: A binary variable indicating that $d_{i,j} \leq S_T^i$.
- $v_{i,j}$: The gain variable resulting from separation $d_{i,j}$.

Figure 3 shows the disjunctive MILP model for the GEO-CAPE scheduling problem. The objective is to maximize the sum of total gains of observations on each pseudo-scene.

[1]The two variables $b_{i_1,j_1;i_2,j_2}$ and $b_{i,j;k}$ have been chosen with the same name because of their similar semantics.

$$\max \sum_{i \in \Psi} \sum_{j \in A_i^*} v_{i,j}$$

$$t_{i_1,j_1} \geq t_{i_2,j_2} + o_{i_2,j_2} - |H| \, b_{i_1,j_1;i_2,j_2} \quad \forall i_1, i_2 \in \Psi, i_1 \neq i_2, C^{i_1} \cap C^{i_2} \neq \emptyset, \forall j_1 \in A_{i_1}, \forall j_2 \in A_{i_2} \tag{1}$$

$$t_{i_2,j_2} \geq t_{i_1,j_1} + o_{i_1,j_1} - |H| \, (1 - b_{i_1,j_1;i_2,j_2}) \quad \forall i_1, i_2 \in \Psi, i_1 \neq i_2, C^{i_1} \cap C^{i_2} \neq \emptyset, \forall j_1 \in A_{i_1}, \forall j_2 \in A_{i_2} \tag{2}$$

$$t_{i,j} \geq \min(C^i) \quad \forall i \in \Psi, \forall j \in A_i \tag{3}$$

$$t_{i,j} \leq \max(C^i) \quad \forall i \in \Psi, \forall j \in A_i \tag{4}$$

$$t_{i,j} > \max(\text{NoObs}^i\,[k])(1 - b_{i,j;k}) \quad \forall i \in \Psi, \forall j \in A_i, \forall k \in \text{NoObs}^i \tag{5}$$

$$t_{i,j} < \min(\text{NoObs}^i\,[k]) b_{i,j;k} + |H| \, (1 - b_{i,j;k}) \quad \forall i \in \Psi, \forall j \in A_i, \forall k \in \text{NoObs}^i \tag{6}$$

$$t_{i,j+1} \geq t_{i,j} + S_B^i \cdot o_{i,j+1} \quad \forall i \in \Psi, \forall j \in A_i^* \tag{7}$$

$$o_{i,j} \geq o_{i,j+1} \quad \forall i \in \Psi, \forall j \in A_i^* \tag{8}$$

$$d_{i,j} \geq 0 \quad \forall i \in \Psi, \forall j \in A_i^* \tag{9}$$

$$d_{i,j} \leq |H| \, o_{i,j+1} \quad \forall i \in \Psi, \forall j \in A_i^* \tag{10}$$

$$d_{i,j} = t_{i,j+1} - t_{i,j} \quad \forall i \in \Psi, \forall j \in A_i^* \tag{11}$$

$$d_{i,j} \leq S_T^i \cdot z_{i,j} + |H| \, (1 - z_{i,j}) \quad \forall i \in \Psi, \forall j \in A_i^* \tag{12}$$

$$d_{i,j} \geq S_T^i \cdot (1 - z_{i,j}) \quad \forall i \in \Psi, \forall j \in A_i^* \tag{13}$$

$$v_{i,j} \leq V_T^i \cdot z_{i,j} \quad \forall i \in \Psi, \forall j \in A_i^* \tag{14}$$

$$v_{i,j} \leq V_T^i \cdot o_{i,j+1} \quad \forall i \in \Psi, \forall j \in A_i^* \tag{15}$$

$$v_{i,j} \leq V_B^i + |H| \, (1 - z_{i,j}) + (d_{i,j} - S_B^i) \cdot \frac{V_T^i - V_B^i}{S_T^i - S_B^i} \quad \forall i \in \Psi, \forall j \in A_i^* \tag{16}$$

Figure 3: A disjunctive MILP model for the GEO-CAPE scheduling problem

Non-overlapping of observations between scenes is formulated by the disjunctive constraints (1) and (2) for each pair of observations belonging to different pseudo-scenes that have some common timeslots. Remark that if one of the two observations is not scheduled (for instance $o_{i_2,j_2} = 0$), the constraints make it possible to select a value for $b_{i_1,j_1;i_2,j_2}$ that permit simultaneity $t_{i_1,j_1} = t_{i_2,j_2}$. Constraints (3) and (4) state the time window for each pseudo-scene i. Disjunctive constraints (5) and (6) forbid observations to occur within non-observability intervals. Constraint (7) guarantees the minimum baseline separation between two consecutive observations when both are executed. Constraint (8) enforces that observation $o_{i,j}$ will be executed if at least j observations of pseudo-scene i are made. Possible values of separation time are stated in constraints (9) and (10), and its expression is defined by constraint (11). Note that constraint (10) enforces that the time value of all absent observations for a given pseudo-scene is equal to the time of the last executed observation. Constraints (12) and (13) are used to calculate $z_{i,j}$ corresponding to its definition. Constraints (14-16) calculate the $v_{i,j}$ values.

Disjunctive MILP evaluation: The performance of this model was evaluated on the GEO-CAPE benchmark (NASA 2016) using CPLEX V12.7. It could solve to optimality 25 out of the 44 instances of the benchmark within the 1 hour time limit (20/22 on the FR instrument, 5/22 on the COEDI one). The disjunctive formulation works well when the non-overlapping constraint between the observations is not the main bottleneck. This is not the case for the instances using the COEDI instrument and for these problems we observed that except for the five ones that are solved to optimality, very poor solutions are produced by the disjunctive MILP (usually worse than the ones of the original time-indexed MILP). Furthermore, the disjunctive formulation is not straightforward and would be hard to adapt to the changing requirements of the real-life application. Even if this disjunctive formulation cannot be considered as an efficient and robust enough candidate for producing good quality schedules, it still can be relaxed to produce tight upper bounds for all problems as described below.

4.2 Upper bounds derived from the disjunctive MILP model

The major source of complexity of the disjunctive MILP model is related with constraints (1) and (2). These non-overlapping constraints can be relaxed by using some form of energetic reasoning (Lopez and Esquirol 1996). If we know an upper bound NbMaxObs on the number of observations that can be scheduled in the time horizon, we can replace constraints (1) and (2) by this relaxation:

$$\sum_{i \in \Psi} \sum_{j \in A_i} o_{i,j} \leq \text{NbMaxObs} \tag{17}$$

For computing such an upper bound NbMaxObs, we adapted the original time-indexed MILP model with the two following changes:

- Change the objective function so as to maximize the number of scheduled observations
- Add a constraint that forbids isolated observations (see proposition 1)[2]

The optimal solution to this adapted formulation is clearly an upper bound on the number of schedulable observations. It turns out that this adapted formulation can be solved to optimality for all instances in reasonable time. The value for NbMaxObs is then used by the relaxed disjunctive MILP consisting of constraints (3-17). These relaxed disjunctive formulations can also all be solved in reasonable time (less than 2 mn), producing the upper bounds shown in Figure 6.

[2]We experimented with a version of the original time-indexed MILP (using the original objective function) with this additional dominance rule but it did not result in significant improvements.

5 CP Optimizer model

In this section we describe a CP Optimizer model for the GEO-CAPE scheduling problem. This model is seldom able to produce optimality proofs but we will show that it consistently provides solutions of excellent quality. Another advantage of this model is that it can be easily adapted to handle the changing requirements of the actual application.

5.1 Basic model

Figure 4 shows a complete CP Optimizer model for the GEO-CAPE scheduling problem written in OPL (Van Hentenryck 1999), version 12.7. This model has some similarities with the CP formulation used in (Frank, Do, and Tran 2016) but it also has some important differences that we will highlight after presenting the model and its results.

```
1  using CP;
2  int   SB = ...; int   ST = ...;
3  float VB = ...; float VT = ...;
4  float A = (VT-VB) / (ST-SB);
5  int n = ...;
6  {int} T[1..n] = ...;
7  int TL = min(i in 1..n, t in T[i]) t;
8  int TU = max(i in 1..n, t in T[i]) t;
9  int m = (TU-TL) div SB;
10
11 pwlFunction V = piecewise{ A->ST; -VT->ST+1; 0 } (SB,VB);
12 stepFunction NoObs[i in 1..n] =
13   stepwise(t in TL-1..TU) { (t in T[i]) -> t+1; 0 };
14
15 dvar interval a [1..n, 1..m+1] optional size 1;
16 dvar interval s [1..n, 1..m]   optional;
17 dvar interval sv[1..n, 1..m]   optional size SB..ST;
18 dvar interval s0[1..n, 1..m]   optional size ST+1..TU;
19
20 maximize sum(i in 1..n, j in 1..m) lengthEval(s[i,j], V);
21 subject to {
22   forall(i in 1..n, j in 1..m+1) {
23     if (j < m+1) {
24       presenceOf(a[i,j+1]) == presenceOf(s[i,j]);
25       startAtStart(a[i,j], s[i,j]);
26       endAtStart(s[i,j], a[i,j+1]);
27       alternative(s[i,j], append(sv[i,j], s0[i,j]));
28       if (j == 1) {
29         presenceOf(a[i,j+1]) == presenceOf(a[i,j]);
30         !presenceOf(s0[i,j]);
31       } else {
32         presenceOf(a[i,j+1])  => presenceOf(a[i,j]);
33         presenceOf(s0[i,j-1]) => presenceOf(sv[i,j]);
34       }
35     }
36     forbidExtent(a[i,j], NoObs[i]);
37   }
38   noOverlap(a);
39 }
```

Figure 4: Complete CP Optimizer model for the GEO-CAPE scheduling problem

The model at line 1 states that it is a CP model. Lines 2-9 read data and compute the value slope (`A`) and an upper bound on the number of observations per scene (`m`). It is to be noted that in the instances of the benchmark, the parameters S_B^i, S_T^i, V_B^i and V_T^i do not depend on the scene i, they are the same for all scenes of the instance. The value gain function `V` of Equation 1 and Figure 1 is defined at Line 11 as a piecewise linear function. Lines 12-13 define, for each scene `i`, a step function `NoObs[i]` with value `0` when the scene is not observable and value 1 otherwise.

Decision variables of the problem are defined in Lines 15-18. Most of the constraints between these variables are illustrated in Figure 5. For a scene `i`, the `j`th observation is modeled by an optional interval variable `a[i,j]` (Line 15). The separation time between the `j`th and `j+1`th observation is represented by an optional interval variable `s[i,j]` (Line 16) that is constrained to start at the start date of `a[i,j]` (Line 25) and end at the start date of `a[i,j+1]` (Line 26). Two situations are possible for this separation time: it either lasts for less than `ST` and in this case it will produce some gain or, it lasts for more and will not produce any gain. These two situations are modeled by two interval variables `sv[i,j]` and `s0[i,j]`: the possible length of `sv[i,j]` is in `[SB,ST]` (Line 17) while the possible length of `s0[i,j]` is greater than `ST` (Line 18). The alternative choice between the two intervals is posted as an `alternative` constraint (Line 27). Constraint at Line 32 states that `a[i,j]` is present if at least `j` observations are made on scene `i` while constraint at Line 24 enforces that the separation intervals `s[i,j]` before the last observation are all present. Constraints at lines 29, 30 and 33 enforce the dominance rule that forbid the presence of isolated observations. Line 33 says that a separation interval that does not produce any gain (`s0[i,j-1]`) must be followed by a separation interval that produces some gain (`sv[i,j]`) as otherwise, observation `a[i,j]` would be isolated. Line 29 says that it is not allowed to have a scene with a unique (necessarily isolated) observation and Line 30 tells that the first observation cannot be isolated. Finally, constraint at Line 36 enforces that if an observation is made, it must be made at a timeslot where the scene is observable and Line 38 ensures that none of the observations are performed at the same time. The objective function (Line 20) is to maximize the total gain, that is the sum of the value of piecewise linear function `V` evaluated on the length of separation intervals `s[i,j]`.

Model evaluation: The performance of this model was evaluated on the GEO-CAPE benchmark using the automatic search of CP Optimizer V12.7 (no parameter change) in similar conditions as the empirical evaluation in the original paper: one hour time limit on a similar (although slightly slower) machine. The instances involve between 932 and 5511 interval variables. Figure 6 shows the average gain obtained by the model of Figure 4 (`Basic CPO`), when run on each instance with 10 different random seeds[3], compared to the gain of the original MILP model proposed in (Frank, Do, and Tran 2016) (`Original MILP`). Except for the FR instrument where all the methods, including the original MILP, find optimal or very good solutions, one sees that the basic CP Optimizer model consistently outperforms the original MILP. On the COEDI instrument, the average optimality

[3]CP Optimizer uses a stochastic search algorithm. We measure the robustness of the engine on an instance by running the search with different random seeds and analyzing the distribution of the objective function value.

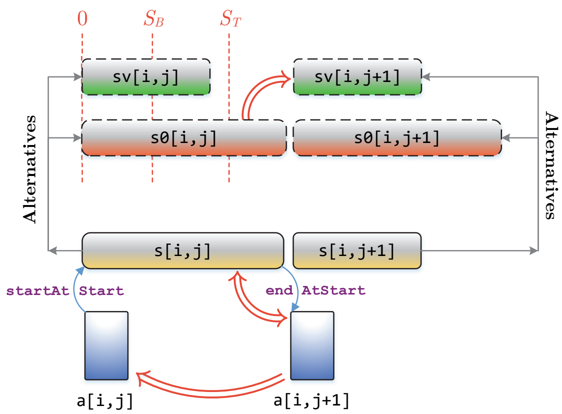

Figure 5: Constraints between consecutive observations in the CP Optimizer model

gap of the basic CP Optimizer model is 3.78% compared to a gap of 18.42% for the original MILP. On this instrument, the improvement is particularly important for the months 4,5,6,8,9,10. Furthermore, the model is robust as the average ratio of the standard deviation of the gain (among the 10 random seeds) to the upper-bound is about 1% only for the COEDI instrument. Another advantage of this model is its capability to quickly produce good quality solutions as illustrated on a representative instance in Figure 7. We tried the same model without enforcing the property of uselessness of isolated observations (lines 29, 30 and 33): the results were slightly worse but still clearly better than the ones of the original time-indexed formulation.

Discussion: In the original paper, a CP formulation was proposed that did not work as well as the original MILP. The basic CP Optimizer model described in the present paper differs from this original CP formulation in that it exploits essential features of the CP Optimizer modeling concepts for scheduling. In particular optional interval variables are used for representing the separation times. Direct constraints can be posted on these optional interval variables instead of composite constraints[4] or constraints à-la MILP involving “big M's”. Looking at the model of Figure 4, the only composite constraints are the ones of Lines 24, 29, 32 and 33. These constraints are logical binary constraints between presence status of interval variables that are efficiently handled by the logical network of CP Optimizer and exploited to perform propagation on conditional domains (start, end) of interval variables even when the presence status is still unfixed (Laborie and Rogerie 2008). The objective function is also directly formulated on the interval variables thanks to the evaluation of the interval length on a piecewise linear function (Line 20); this type of function is efficiently exploited by the temporal linear relaxation used to guide the search (Laborie and Rogerie 2016). One last difference[5] is that the original CP formulation seems to over-constrain the problem by enforcing a single block of observations per scene (constraint 16 in the model) but it does not prevent isolated observation. On the contrary, our formulation allows for more than one block but, as a legal dominance rule, prevents isolated observations. Compared to the MILP models (both the original model and the disjunctive one described in this paper), the CP formulation is more flexible due to the expressivity of CP and can for instance be easily extended to handle observations with non-unit durations (change the unit size at Line 15) or observations that can be performed according to several alternative modes (create some alternative between the observations variables `a[i,j]` and a set of additional optional variables `am[i,j,k]` for each mode `k`). Sequence-dependent setup times between observations could easily be modeled by adding a transition matrix in the `noOverlap` constraint at line 38. In the context of rescheduling, freezing some decisions of the current schedule (like keeping some scheduled observations and preventing the degradation of their gain) is easy to model as additional constraints.

5.2 Advanced model

While the basic CP Optimizer model of Figure 4 performs remarkably well compared to the original MILP, we investigated the following additional improvements.

Decomposition in pseudo-scenes: Following proposition 3, we automatically decompose the scenes of the original problem into a set of pseudo-scenes. This decomposition reduces the length of the implication chains (produced by the constraints at Line 32) which is an element contributing to the complexity of the model.

Maximal number of observations per pseudo-scene: For simplicity reasons, the basic model computes a maximal number of observations that is global to all the scenes (Line 9). In the advanced model, this maximal number of observation is computed individually for each pseudo-scene. This results in a smaller number of defined interval variables (about half less variables in the benchmark).

A two-stage resolution approach: The objective function of the basic model (Line 20) clearly is not convex due to the shape of function V. The temporal linear relaxation will automatically convexify it (Laborie and Rogerie 2016) but this will result in some loose relaxations that may impact the performance of the search. Instead, the advanced CP Optimizer model solves the problem in two stages:

1. In a first stage, a surrogate *convex* objective function is used in place of the original objective. Proposition 2 suggests that the total length of the productive separation times (so the total length of the trains of observations) can be considered as a good approximation of the gain value (specially when β_i is small). So the original objective function is replaced by `sum(i,j) lengthOf(sv[i,j])`.

[4]Composite constraints are logical or arithmetical combinations of constraints like $ct1 \vee ct2$. In CP these constraints are known to result in quite loose constraint propagation unless specifically handled by the CP engine (Barták, Salido, and Rossi 2010).

[5]The impact of this last difference in terms of performance is less significant according to our experiments

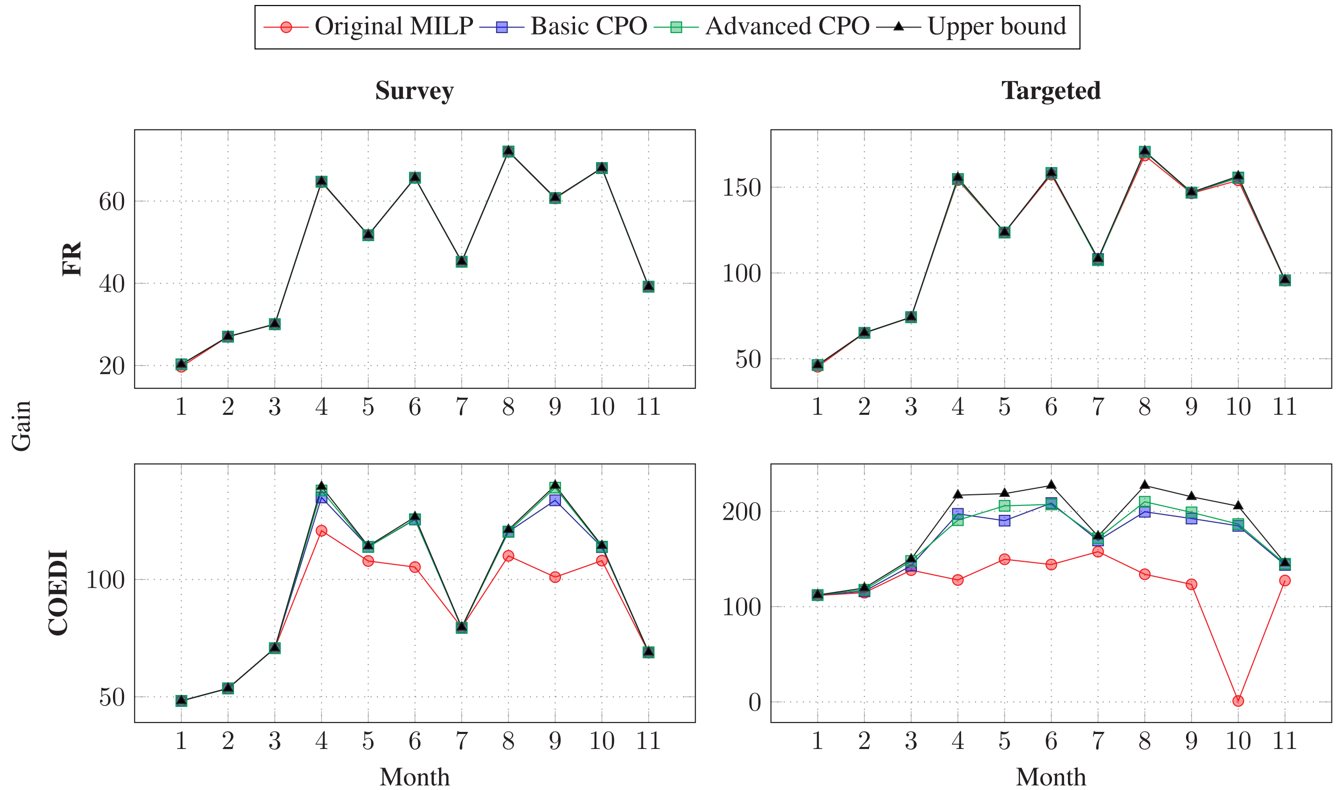

Figure 6: Schedule quality comparisons of the different approaches for Survey (left graphs) and Targeted science (right graphs) instances. Top graphs are instances using the FR instrument while bottom graphs are the ones using COEDI.

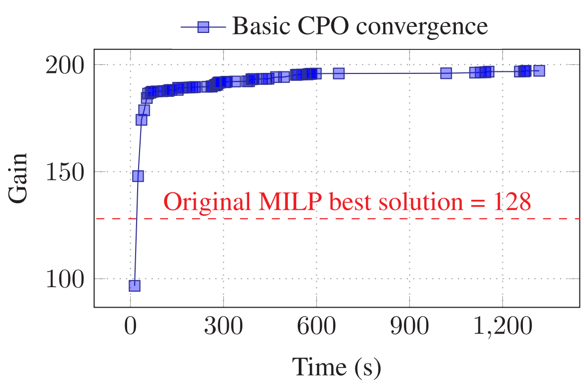

Figure 7: Typical convergence of the basic CP Optimizer model on an instance (here: *COEDI Targeted, Month=4*)

2. The second stage uses the original objective function. The best solution found in the first stage is re-injected as warm start into the second stage by using the CP Optimizer concept of a *starting point* (IBM 2016).

Advanced model evaluation: In the evaluation of the advanced approach we split the one hour time limit into 30 minutes for the first stage and 30 minutes for the second. Both stages are solved using the automatic search of CP Optimizer V12.7. In a preliminary study, we noticed that all three modifications of the basic CP model contribute to the performance improvements. We summarize in Figure 6 the results using the conjunction of the three modifications, averaged over 10 random seeds. We see that the advanced CP Optimizer approach generally improves the results of the basic CP Optimizer model. On the COEDI instrument, the average optimality gap of the advanced CPO model is 2.72% compared to a gap of 3.78% for the basic model and 18.42% for the original MILP. Globally, this advanced model finds optimal or near-optimal solutions (optimality gap smaller than 1%) for about 80% of the instances (35 out of 44).

Schedule efficiency: In the original paper, besides the total gain, the authors also analyze the solutions in terms of *efficiency* which is defined as the ratio between the number of scheduled observations and the number of available timeslots. This indicator is particularly relevant in case we forbid non-isolated observations in the solution as these observations do not bring any gain[6]. Intuitively, good quality solutions will tend to schedule as many non-isolated obser-

[6]We conjecture that this is a reason why some solutions of the original MILP with a good efficiency turn out to be rather poor in terms of gain as they contain many isolated observations. For example the efficiency of *COEDI Targeted Month=6* exceeds 90% for a gain of 144.2 whereas we can produce a much better solution of gain 209.3.

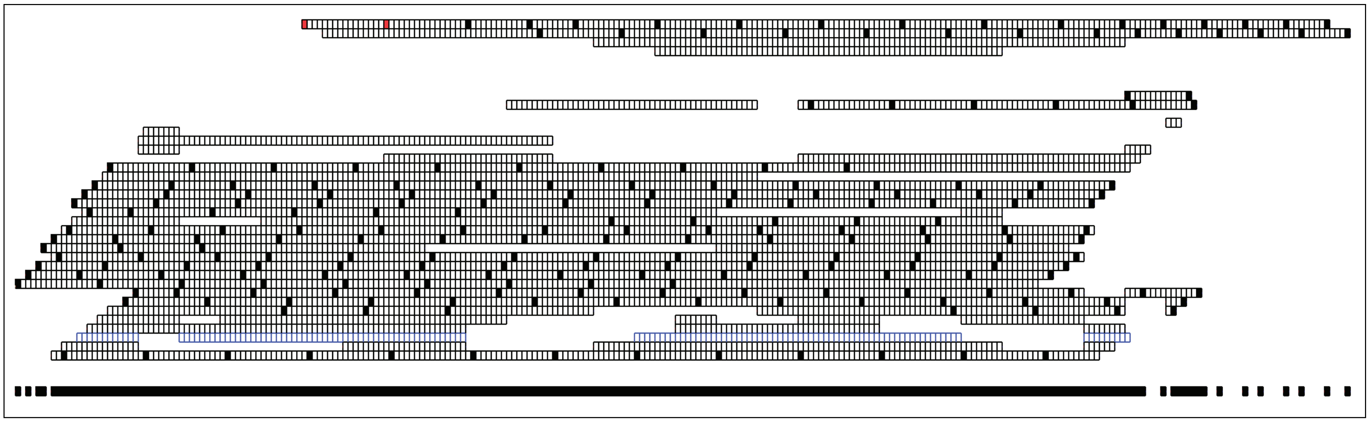

Figure 8: Example of a solution produced by CP Optimizer for instance *COEDI Targeted, Month=9*

vations as possible. Interestingly, we know for each instance the maximal number of schedulable non-isolated observations (see subsection on upper bounds) so we can compute the ratio between the number of scheduled observations and this theoretical maximum. For instance on the solution of Figure 8, 233 observations are scheduled whereas the theoretical maximal number of schedulable non-isolated observations is 235 and the number of timeslots is 263. This means that most of the timeslots where no observation is performed (last line of the figure) are unavoidable. For instance, it is not possible to schedule two observations at the two first timeslots of the schedule: they belong to a unique scene and the two observations on this scene would not respect the minimal separation time. Similarly, the gaps in the very end of the schedule are due to the fact there are only two scenes in this region. For all the instances of the benchmark, our solutions always schedule more than 99% of the maximal number of schedulable non-isolated observations.

6 Conclusion

This paper presents new results on the GEO-CAPE Observation Scheduling Problem introduced in (Frank, Do, and Tran 2016). In order to estimate and compare the quality of existing and new solutions, we first presented a disjunctive MILP model. This formulation solves more than half of the instances of the benchmark and can be relaxed to provide interesting upper bounds for the other problems. We then described a concise and efficient CP formulation. This formulation is a very good example of how to leverage the CP Optimizer concepts for scheduling, in particular the central one of *optional interval variables*. It results in a model without reified constraints that gives the CP engine a direct grip on the structure of the problem. A slightly improved version of this basic CP model is able to produce optimal or near-optimal solutions (optimality gap smaller than 1%) for about 80% of the instances. On the most challenging family of problems (COEDI instrument), the average gap is reduced to 2.72% compared to 18.42% for the existing solutions. Interestingly, because of the better quality schedules produced for the COEDI instrument, the difference of performance between the two instruments reported in (Frank, Do, and Tran

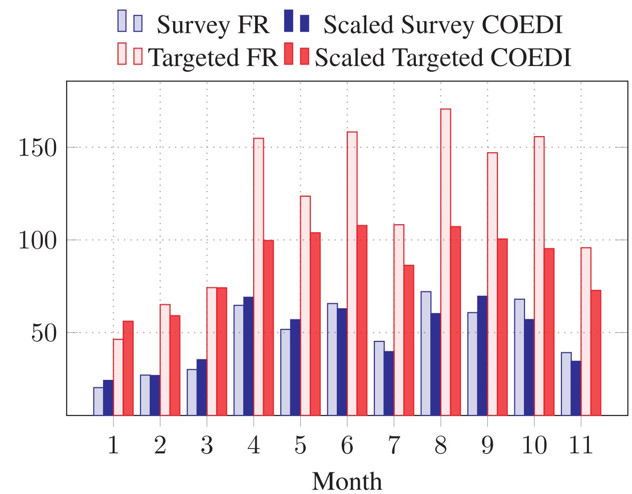

Figure 9: Comparison of schedule scores for FR and COEDI instruments

2016) is slightly re-equilibrated in favor of COEDI. Even if FR still dominates COEDI for the Targeted science problems, we see in Figure 9 that the two instruments are equivalent for the Survey science ones.

An advantage of the CP model is its flexibility to cope with the changing requirements and possible changes of the scope of the application: different observation durations, multi-mode (Weglarz et al. 2011), sequence-dependent setup times (Allahverdi et al. 2008), rescheduling, *etc.* In the future it would be interesting to study these developments. Another extension would be to improve the upper bounds by considering the interactions between the separation times and the non-overlapping of observations, for instance it is not possible to have more than S_T trains of observations simultaneously executing in the schedule. We conjecture that a non-negligible part of the optimality gap is due to the weakness of the upper bounds and that the solutions of the CP Optimizer model are closer to the optimal value than the current gap would suggest.

Acknowledgements

We are thankful to Jeremy Frank, Minh Do and Tony Tran who made their benchmark available and kindly helped us in using it.

References

Allahverdi, A.; Ng, C.; Cheng, T.; and Kovalyov, M. Y. 2008. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research* 187:985–1032.

Barták, R.; Salido, M. A.; and Rossi, F. 2010. Constraint satisfaction techniques in planning and scheduling. *Journal of Intelligent Manufacturing* 21:5–15.

Frank, J.; Do, M.; and Tran, T. T. 2016. Scheduling Ocean Color Observations for a GEO-Stationary Satellite. In *Proceedings of the 26th International Conference on Automated Planning and Scheduling (ICAPS-2016)*.

IBM. 2016. ILOG CPLEX Optimization Studio 12.7.0: CP Optimizer Online Documentation. Available at `http://ibm.biz/COS127Documentation`.

Jouglet, A., and Carlier, J. 2011. Dominance rules in combinatorial optimization problems. *European Journal of Operational Research* 212:433–444.

Ku, W.-Y., and Beck, J. C. 2016. Mixed Integer Programming models for job shop scheduling: A computational analysis. *Computers & Operations Research*.

Laborie, P., and Rogerie, J. 2008. Reasoning with Conditional Time-intervals. In *Proc. 21th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2008)*.

Laborie, P., and Rogerie, J. 2016. Temporal Linear Relaxation in IBM ILOG CP Optimizer. *Journal of Scheduling* 19(4):391–400.

Laborie, P. 2009. IBM ILOG CP Optimizer for Detailed Scheduling Illustrated on Three Problems. In *Proc. CPAIOR 2009*.

Lopez, P., and Esquirol, P. 1996. Consistency enforcing in scheduling: A general formulation based on energetic reasoning. In *5th International Workshop on Project Management and Scheduling (PMS'96)*.

NASA. 2016. GEO-CAPE Public Distribution. Available at `https://ti.arc.nasa.gov/tech/asr/planning-and-scheduling/pastprojects/`.

Van Hentenryck, P. 1999. *The OPL Optimization Programming Language*. MIT Press.

Weglarz, J.; Józefowska, J.; Mika, M.; and Waligóra, G. 2011. Project scheduling with finite or infinite number of activity processing modes A survey. *European Journal of Operational Research* 208:177–205.