

A Mathematical Programming Algorithm for Planning and Scheduling an Earth Observing SAR Constellation

Nicola Bianchessi and Giovanni Righini

Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
Via Bramante 65, 26013 Crema, Italy
{bianchessi,righini}@dti.unimi.it

Abstract

We consider the planning and scheduling problem of the COSMO-SkyMed constellation for the observation of the Earth. We define a simplified problem which only takes into account the most binding constraints of the real problem. We present a Lagrangean relaxation algorithm that provides upper bounds and we show that it can also be used to guide an existing heuristic algorithm to find better solutions to the real problem.

Introduction

This work stems from the development of planning and scheduling algorithms for the COSMO-SkyMed constellation of ASI - Agenzia Spaziale Italiana. The *COSMO-SkyMed* constellation for the observation of the Earth is made of four satellites equipped with SAR (Synthetic Aperture Radar) instruments. Owing to the SAR technology there is no difference between day and night observations and the full orbit is available for image acquisition. Since the system is oversubscribed the planning and scheduling problem consists in selecting a feasible subset of images to be taken by each satellite over the planning horizon with the objective of providing maximum satisfaction to the customers' requests.

The literature on planning and scheduling problems related to Earth observing satellites is rich but most contributions consider a single satellite or a single orbit. For a comprehensive survey we refer the reader to (Bianchessi 2006). As far as constellations of SAR satellites are concerned a heuristic algorithm capable of solving the planning and scheduling problem for a huge number of image requests per day was presented in (Bianchessi *et al.* 2004). It is a greedy constructive algorithm enhanced with backtracking and look-ahead capabilities, developed for the *COSMO-SkyMed* constellation. Another recent contribution on the same stream is the one of De Florio, Neff, and Zehetbauer (2005), who present two greedy algorithms for a very similar problem. A comparison of local search techniques was illustrated in (Globus *et al.* 2004).

The aim of designing a mathematical programming algorithm based on Lagrangean relaxation is twofold: on one

hand we look for upper bounds to the optimal value, in order to evaluate the performance of the existing heuristic algorithm; on the other hand, we try to identify the structure of "almost feasible solutions" that can guide the above mentioned heuristic algorithm to find better solutions than those computed from scratch.

With respect to the problem solved in (Bianchessi *et al.* 2004) we consider here a simplified model still capturing the most critical combinatorial aspects and the most binding constraints, that is those on memory capacity, set-up times and operational profiles.

For each requested image we assume that some acquisition possibilities (DTOs, i.e. Data Take Opportunities) have been identified and each of them is associated with a given time window.

Since the SAR equipment can work in different operating modes (such as WIDEFIELD and SPOTLIGHT) and the satellites can fly in different positions, set-up operations are also required. A change in the set-up requires that a minimum transition time elapse between two consecutive acquisitions.

Acquired images are stored on board until they can be downloaded to one or more ground stations: the recording capacity on board cannot be exceeded. The connection between a satellite and a ground station takes place only within pre-specified time windows called DLOs (Down Link Opportunities).

Finally the consumption of energy for each given time period cannot exceed a certain threshold. These constraints are represented in our case by a set of operational profiles. In every orbit the time spent to acquire WIDEFIELD images must be less than or equal to a given limit \bar{T} and the total number of SPOTLIGHT images taken must be less than or equal to a given limit \bar{S} ; these limits define *nominal profiles*. In addition for every satellite there is some possibility of violating the nominal profile constraints in a controlled way during some so called *peak orbits*, in which a limit \bar{P} is imposed on the overall acquisition time.

Formulation

Let us introduce some notation that we use in the remainder. K indicates the set of satellites and W the set of required images. For each image $w \in W$, D_w is the set of DTOs associated with the image. Let $D = \cup_{w \in W} D_w$ be the set

of all DTOs. For each DTO $i \in D$, w_i indicates the corresponding image and $[a_i, b_i]$ is the time interval necessary to take it.

Graph representation. We model the simplified problem by means of a multi-commodity flow formulation based on the unified framework described in (Desaulniers *et al.* 1998). This kind of formulation allows to model combinatorial optimization problems, where a set of *commodities* (vehicles, crews, ...) have to perform *tasks* by traveling along paths in a physical or logical network, with the aim to optimize an objective function, while satisfying *local* and *global* constraints. Local constraints concern the construction of feasible paths for each commodity, whereas global constraints define the feasibility of the overall set of paths traveled by the commodities. Tasks can be associated with nodes or arcs on the network; for our problem we adopt a task-on-arc formulation. A *commodity* and a *task* represent a satellite and an image respectively.

A directed acyclic digraph $G^k = (V^k, A^k)$ is defined for each commodity $k \in K$. In the remainder we drop the superscript k when this does not provoke ambiguity. With the task-on-arc representation the nodes in V represent the DTOs for the generic satellite. Two special nodes o and d represent the source and the sink for the commodity. Each other node i represents a DTO for that satellite and all nodes are numbered following the chronological order of the corresponding DTOs. When the commodity visits node i , this means that the satellite set-up is the one needed to acquire DTO i . For each pair of nodes (i, j) the arc (i, j) belongs to A if $b_i + t_{ij} \leq a_j$, where t_{ij} is the set-up time needed to take image j after image i . If the commodity flows along arc (i, j) this means that the satellite takes DTO j after DTO i . An arc (i, d) is also inserted in A for each node i . Satellite unavailability periods can also be easily handled in the construction of the graph. A detailed description of the graph construction procedure, including reduction techniques, is reported in (Bianchessi 2006).

A profit v_{ij} is associated with each arc (i, j) : it represents the value associated with image j . Furthermore each arc (i, j) is characterized by resource consumptions, related to the constraints on memory and operational profiles.

For each arc (i, j) the memory consumption is given by $m_{ij} = m_{w_j} - \delta_{ij}$, where m_{w_j} is the size of image j and δ_{ij} is the amount of data that can be transmitted to ground stations in the time interval $[b_i, b_j]$. If during this time interval no DLO is available for the satellite, then $\delta_{ij} = 0$. To enforce memory capacity constraints we define a variable representing the memory consumption along the path followed by the commodity. Let Q_i be the amount of resource consumption accumulated at node i along the path. The constraint on memory capacity can be expressed by $Q_i \leq M$ for each node i visited along the path, where M is the capacity of the memory on board. For each arc (i, j) along the path we have $Q_j = \max\{0, Q_i + m_{ij}\}$.

Analogously we define ρ_{ij} and σ_{ij} respectively as the consumption of time in WIDEFIELD observations and the number of SPOTLIGHT images taken. For each arc (i, j)

if DTO j is associated with a WIDEFIELD image, then $\rho_{ij} = b_j - a_j$ and $\sigma_{ij} = 0$; on the other hand if DTO j is associated with a SPOTLIGHT image, then $\rho_{ij} = 0$ and $\sigma_{ij} = 1$. We check the nominal profiles only in a discrete number of orbits: for each commodity we fix a number Z of *train of orbits* to be considered and we define them in the following way. For each $z \in \{1, \dots, Z\}$ we split the time horizon in a sequence of *consecutive* time windows one orbit large starting at time $\frac{O}{Z}(z-1)$, where O is the duration of an orbit. Therefore each time instant belongs to exactly Z orbits, one for each train of orbits. To check the constraints on the operational profiles we define a set of three resources for each commodity and for each train of orbits $z \in \{1, \dots, Z\}$: T_i^z represents the time spent in acquiring WIDEFIELD images up to node i , S_i^z represents the number of SPOTLIGHT images taken up to node i and P_i^z represents the consumption of the additional time in peak orbits up to node i . In our attempt to compute upper bounds the constraint on the use of additional time allowed by peak orbits is relaxed: the additional time is not restricted to be available in a time window of given and limited width but it can be consumed whenever needed. For each arc $(i, j) \in A$ and for each $z \in \{1, \dots, Z\}$ the value of the resource variables T , S and P are updated by means of *resource extension functions* depending on the weights ρ and σ of the arcs traversed. More details on the definition of these functions can be found in (Bianchessi 2006). A path is feasible only if $T_i^z \leq \bar{T}$, $S_i^z \leq \bar{S}$ and $P_i^z \leq \bar{P} \forall z \in \{1, \dots, Z\}$ and for all nodes i belonging to the path. In other words a path is feasible only if a feasible amount of each resource has been consumed at each node along it.

A mathematical programming model. For each satellite $k \in K$, we use binary variables x_{ij}^k with $(i, j) \in A^k$ as flow variables which are equal to 1 for the arcs in the path of commodity k and 0 otherwise. An integer programming formulation of our planning and scheduling problem is the following:

$$\max \sum_{k \in K} \sum_{(i,j) \in A^k} v_{ij}^k x_{ij}^k \quad (1)$$

$$\text{s.t.} \sum_{k \in K} \sum_{(i,j) \in A^k | w_j = \bar{w}} x_{ij}^k \leq 1 \quad \forall \bar{w} \in W \quad (2)$$

$$\sum_{j \in \Gamma_o^k} x_{oj}^k = 1 \quad \forall k \in K \quad (3)$$

$$\sum_{j \in \Gamma_i^k} x_{ij}^k - \sum_{j \in \Psi_i^k} x_{ji}^k = 0 \quad \forall k \in K, \forall i \in V^k \quad (4)$$

$$\sum_{j \in \Psi_d^k} x_{jd}^k = 1 \quad \forall k \in K \quad (5)$$

$$x^k \in X_k \quad \forall k \in K \quad (6)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall k \in K, \forall (i, j) \in A^k \quad (7)$$

where Γ_i^k and Ψ_i^k represent respectively the set of successor and predecessor nodes of node i in G^k . The objective function (1) asks for the maximization of the overall profit of the

arcs traversed. Constraints (2) state that each request must be satisfied at most once. Constraints (3)–(5) are the flow conservation constraints for paths originating at the source nodes and terminating at the sink nodes. With X_k we indicate the set of feasible solutions with respect to resource constraints on memory and operational profiles. Constraints (7) impose binary values to the flow variables.

Constraints (3)–(7) are separable by satellite and they define a *Shortest Path Problem with Resource Constraints* (SP-PRC) for each commodity $k \in K$. Remarkably the resources in this problem are not only consumed but also regenerated: the available memory is increased every time the satellite passes over a ground station and the operational profile resources are regenerated for every new orbit in each train of orbits.

The combinatorial structure of the formulation above can be exploited by Lagrangean relaxation in order to decompose the problem, as shown in the next section.

Lagrangean relaxation

Let us recall some basic concepts about Lagrangean relaxation. Consider the following integer problem:

$$(IP) \quad \begin{aligned} z &= \max \mathbf{c}^T \mathbf{x} \\ A\mathbf{x} &\leq \mathbf{b} \\ D\mathbf{x} &\leq \mathbf{d} \\ \mathbf{x} &\in \mathbb{Z}_+^n \end{aligned}$$

where A is an $l \times n$ matrix, D is an $m \times n$ matrix, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^l$ and $\mathbf{d} \in \mathbb{R}^m$. It may happen that removing “complicating constraints” $D\mathbf{x} \leq \mathbf{d}$, the resulting integer problem becomes easier to solve or can be decomposed into a set of independently solvable easier subproblems. For any value of $\mu = (\mu_1, \dots, \mu_m) \geq \mathbf{0}$, the problem

$$(IP(\mu)) \quad \begin{aligned} z(\mu) &= \max \mathbf{c}^T \mathbf{x} + \mu(\mathbf{d} - D\mathbf{x}) \\ A\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\in \mathbb{Z}_+^n \end{aligned}$$

is the *Lagrangean relaxation* of IP with parameter μ . In $(IP(\mu))$ the complicating constraints are handled by penalty terms in the objective function. The components of μ are the *Lagrangean multipliers* associated with the constraints $D\mathbf{x} \leq \mathbf{d}$. Solving $(IP(\mu))$, we obtain a valid upper bound for the original problem. In order to find the best upper bound we need to solve the *Lagrangean dual* problem defined as:

$$z_{LD} = \min\{z(\mu) \mid \mu \geq \mathbf{0}\}.$$

In our case, if constraints (2), which link the plans of different satellites, are relaxed, the problem can be decomposed into K independent subproblems, one for each satellite. Lagrangean multipliers $\mu_w \geq 0$ with $w \in W$ are associated with the relaxed constraints. For given values of μ the objective function of the Lagrangean relaxation $LR(\mu)$ is

$$z(\mu) = \sum_{k \in K} z^k + \sum_{w \in W} \mu_w$$

where, for each commodity $k \in K$, z^k is the optimal value of the subproblem

$$\max \sum_{(i,j) \in A^k} (v_{ij}^k - \mu_{w_j}) x_{ij}^k \quad (8)$$

$$\text{s.t.} \sum_{j \in \Gamma_o^k} x_{oj}^k = 1 \quad (9)$$

$$\sum_{j \in \Gamma_i^k} x_{ij}^k - \sum_{j \in \Psi_i^k} x_{ji}^k = 0 \quad \forall i \in V^k \quad (10)$$

$$\sum_{j \in \Psi_d^k} x_{jd}^k = 1 \quad (11)$$

$$x^k \in X_k \quad (12)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in A^k \quad (13)$$

that is a SPPRC. In the remainder we describe the algorithms for solving the Lagrangean relaxation, that is the SPPRC instances, and the Lagrangean dual problem.

A dynamic programming algorithm. For each commodity $k \in K$ the SPPRC is defined on the acyclic digraph $G^k = (V^k, A^k)$ constructed as described above. Superscript k is dropped in the remainder, since each satellite is treated separately.

Dynamic programming solution approaches for the SP-PRC build paths from the source node of the graph, by extending them from each node to every feasible successor. Their efficiency depends on the ability to identify and discard dominated paths, that is paths which cannot be extended to optimal paths. With each feasible path from the origin o to a generic node i we associate a label (Q, T, S, P, v, i) , where the first four quantities represent the consumption of resources, v is the accumulated value of the path and i is the last node reached. The labels are iteratively updated along each path by means of the resource extension functions. Let $(Q^1, T^1, S^1, P^1, v^1, i)$ and $(Q^2, T^2, S^2, P^2, v^2, i)$ be two different labels corresponding to two paths from o to a same node i . The first label *dominates* the second one only if $Q^1 \leq Q^2$, $T^1 \leq T^2$, $S^1 \leq S^2$, $P^1 \leq P^2$, and $v^1 \geq v^2$. A label is called *efficient* if no other label dominates it. The maximum profit path from o to d satisfying the resource constraints is obtained directly from the set of the efficient labels associated with node d .

We implemented a *label setting* algorithm. The labels chosen to be extended are kept without modification until the end of the labeling process. The profit of the arcs $v_{ij} - \mu_{w_j}$ may be negative due to the effect of the dual variables; however, for each commodity $k \in K$ the graph G^k is acyclic. Therefore the difficulty of the problem is not due to the presence of negative cost cycles but to the number of (regenerated) resources, which in turn yields a large number of non-dominated labels associated with each node. The efficiency of the dynamic programming algorithm heavily relies upon the data structures used to represent the set of labels and to perform the dominance test. We implemented the set of the efficient labels associated with each node i by

a three-dimensional matrix M of *buckets* with $m \times n \times p$ elements, where $m = \lceil \frac{\bar{T}}{\alpha} \rceil$, $n = \lceil \frac{\bar{S}}{\beta} \rceil$, $p = \lceil \frac{\bar{P}}{\gamma} \rceil$ and α, β, γ are parameters suitably chosen. Each element of the matrix is a bucket, represented by a doubly linked list in which labels are kept in non-increasing order according to their profit. With each matrix element we also associate the minimum and the maximum profit of the labels in the corresponding list. Thanks to this implementation, when a new label is created, we can skip with a simple test a rather large number of dominance tests, saving a significant amount of computing time.

Solving the Lagrangean dual problem. We solve the Lagrangean dual problem in an approximate way by applying the well-known *subgradient* method (Held, Wolfe, and Crowder 1974). At each iteration t the algorithm computes the upper bound $z(\mu^t)$ and records the best value found so far. Then it adjusts the Lagrangean multipliers by moving in the dual space from the current point μ^t in the direction opposite to a subgradient of $z(\mu)$ computed in μ^t .

The algorithm has been implemented in the standard way described in (Beasley 1992), where the rules to update the step length at each iteration are also explained. At each iteration t we also search for a feasible solution and a corresponding lower bound to each subproblem by removing from the optimal solution of $LR(\mu^t)$ all the image acquisitions that lead to a violation of constraints (2).

Almost feasible solutions. By “almost feasible solution” we mean a solution in which some of the constraints of the original planning and scheduling problem can be violated. These constraints may include for instance user quotas, split requests, deadlines, priorities and other constraints on acquisition and transmission. The idea is that almost feasible solutions include a set of data take operations which are moderately conflicting due to the constraints not taken into account in the simplified version of the problem. If a heuristic algorithm is executed on this restricted set of requests, instead of the whole set of requests, it may possibly find a good feasible solution, whose structure resembles the one of the almost feasible solution from which the requests subset has been generated.

To compute almost feasible solutions, we consider the problem (1)-(7) but we take into account only nominal operation profiles in the SPPRC arising for each commodity $k \in K$. We consider one satellite at a time in random order. For each given satellite k we do not consider the images already taken by previously considered satellites. Then we solve the resulting SPPRC by means of a modified version of the dynamic programming algorithm described above that prevents the multiple acquisitions of an image along a path. At the end of this process we obtain a solution made by a path for each commodity.

Then we apply a post-processing procedure for each satellite to exploit the possibility to perform peak orbits for each satellite. For each acquired DTO i listed in the path we consider a time window one orbit large starting at a_i and we

Table 1: Test instances: commodity network size

Inst.	$ W $	SP	WF	$ V $	$ A $
1	1327	1064	263	1573.50	158476.50
2	1000	249	751	1192.75	220034.75
3	2000	369	1631	3977.75	544543.50
4	1000	198	802	2206.00	243825.00
5	1600	319	1281	3432.25	587219.75

evaluate the increase in the solution value that can be obtained by performing new acquisitions within that time window. The new acquisitions are selected by considering sequentially all DTOs that can be taken by the satellite during the orbit: a DTO can be taken if it is compatible with those already listed in the path, and it does not lead to violations of the peak profile constraints and it is associated with an image not yet acquired. The orbit yielding the greatest increase in the solution value is selected as a peak orbit and the path is modified accordingly. We proceed in a similar way to identify triplets of peak orbits, which are subject to special operational profile constraints, similar to those imposed on peak orbits.

The solution obtained is feasible for the simplified problem considered here, but it may be infeasible for the real planning and scheduling problem of a SAR constellation like COSMO-SkyMed. However the subset of DTOs included in this solution can be used as a guide for a heuristic, which takes into account all the problem constraints (including use of shared resources on board, user quotas, split requests, deadlines and priorities), such as the algorithm described in (Bianchessi *et al.* 2004).

Computational results

The algorithms described here have been tested on five problem instances provided by Space Software Italia. Each of them concerns four satellites performing about 15 orbits in a time horizon of one day. In these instances each image has the same profit, so that the objective function reduces to the number of images taken.

In Table 1 we summarize the main characteristics of the instances: $|W|$ is the cardinality of the image set; the next two columns indicate the number of SPOTLIGHT and WIDEFIELD images. The average size of the commodity networks generated from each instance is reported in last two columns: $|V|$ is the average number of nodes and $|A|$ is the average number of arcs.

Experiments have been done on a 1.6 GHz Pentium IV PC. For each instance we have found upper bounds by solving different Lagrangean relaxations. In relaxation $LR^{(1)}(\mu)$ we consider as a resource only the memory and we disregard the operational profiles. In relaxation $LR^{(2)}(\mu)$ we consider only operational profile resources, with only one train of orbits for each commodity. In relaxation $LR^{(3)}(\mu)$ we allow each discrete orbit to be a peak orbit. In relaxation $LR^{(4)}(\mu)$ WIDEFIELD images are com-

Table 2: Upper bounds

Instance	$z_{LD}^{(1)}$	$z_{LD}^{(2)}$	$z_{LD}^{(3)}$	$z_{LD}^{(4)}$
1	898.50	673.94	895.48	954.49
2	703.50	876.62	701.40	821.56
3	2472.97	-	1996.07	-
4	1017.99	1456.48	1002.72	1505.50
5	2085.12	-	1589.84	2164.00

Table 3: Upper bounds, $z_{LD}^{(3)}$, multiple orbit sequences

Instance	$Z = 1$	$Z = 2$	$Z = 3$
1	895.48	894.79	892.81
2	701.40	701.47	700.70
3	1996.07	2328.28	3022.50
4	1002.72	1214.51	1649.00
5	1589.84	2261.00	2260.00

bined linearly with SPOTLIGHT images in the evaluation of nominal profiles. We solve the Lagrangean dual associated with the four mentioned relaxations with a maximum CPU time of 6 hours.

Dual bounds. For each instance, the upper bound z_{LD} of each of the four relaxations is reported in Table 2 and the tightest upper bound is shown in bold.

The upper bounds found by considering operational profiles are systematically tighter than those found by considering the memory resource: the percentage of improvement ranges from 0.14% to 33.32%. This confirms that the operational profile constraints are binding the performance of the constellation. When increasing the number of resources, the number of labels per node in the dynamic programming algorithm increases exponentially. Hence more time is required to solve the SPPRC associated with each commodity and fewer iterations can be performed by the subgradient algorithm within the allotted time. In particular, due to the large size of the instances, only for relaxation $LR^{(3)}(\mu)$ we have been able to approximate the optimal value of the associated Lagrangean dual. This is why the value listed in column $z_{LD}^{(3)}$ are in general better than those reported in columns $z_{LD}^{(2)}$ and $z_{LD}^{(4)}$.

We have tried to improve the values obtained from relaxation $LR^{(3)}(\mu)$ by considering a number of train of orbits equal to 2 and 3. Results are reported in Table 3. Not surprisingly, only for the smallest instances we have been able to obtain slightly better upper bounds. A larger number of orbits gives rise to an increase in the number of resources to consider with the same drawbacks cited above.

Improved heuristic. We have computed lower bounds by guiding the heuristic algorithm presented in (Bianchessi *et al.* 2004) by means of almost feasible solutions. For each

Table 4: Primal bounds

Instance	$z^{(G)}$	$z^{(S)}$	gap(%)
1	548	513	6.82
2	644	588	9.52
3	1384	1506	-8.10
4	783	822	-4.74
5	1033	1034	-0.09

instance we report in column $z^{(G)}$ of Table 4 the results obtained by running the planning and scheduling algorithm on the subset of DTOs taken in almost feasible solutions. Column $z^{(S)}$ contains the results computed by the same algorithm considering all the requests. In the last column of the table we report the gap between the two values.

For the first two instances we could improve the solution respectively by 6.82% and 9.52%, whereas for the other instances the quality of the solutions computed from scratch is better. This suggests that the choice of the acquisitions to schedule becomes less and less important when the number of possible acquisitions grows. Since the time needed to compute the almost feasible solutions for instances 1 and 2 was less than two minutes, this approach represents a valid alternative to solve relatively small instances.

In these instances all requests had the same value; if the observation requests may have different values associated and the objective is to maximize the overall value and not merely the number of requests satisfied, then the difference in performance is likely to be even more significantly in favor of a mathematical programming approach also on large instances.

References

- Beasley, J. E. 1992. Lagrangean relaxation. Technical report, The Management School, Imperial College, London.
- Bianchessi, N.; Piuri, V.; Righini, G.; Roveri, M.; Laneve, G.; and Zigrino, A. 2004. An optimization approach to the planning of earth observing satellites. In *Proceedings of the Fourth International Workshop on Planning and Scheduling for Space (ESA - ESOC)*. Darmstadt, Germany.
- Bianchessi, N. 2006. *Planning and Scheduling Problems for Earth Observation Satellites: Models and Algorithms*. Ph.D. Dissertation, Dipartimento di Tecnologie dell'Informazione, Università degli Studi di Milano.
- DeFlorio, S.; Neff, T.; and Zehetbauer, T. 2005. Optimal operations planning for sar satellite constellations in low earth orbit. In *Proceedings of the Sixth International Symposium on Reducing the Costs of Spacecraft Ground Systems and Operations (ESA - ESOC)*. Darmstadt, Germany.
- Desaulniers, G.; Desrosiers, J.; Ioachim, I.; Solomon, M. M.; Soumis, F.; and Villeneuve, D. 1998. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In Crainic, T. G., and Laporte, G., eds., *Fleet Management and Logistics*. Kluwer, Norwell, MA. 57–93.

Globus, A.; Crawford, J.; Lohn, J.; and Pryor, A. 2004. A comparison of techniques for scheduling earth observing satellites. In *Proceedings of the Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAA-04)*. Sponsored by the American Association for Artificial Intelligence, San Jose, California.

Held, M.; Wolfe, P.; and Crowder, H. P. 1974. Validation of subgradient optimization. *Mathematical Programming* 6:62–88.