

A Tabu Search Algorithm for Satellite Imaging Scheduling*

Wei-Chen Lin

Department of Electrical Engineering
National Taiwan University
Taipei, 106, TAIWAN
byron@ac.ee.ntu.edu.tw

Da-Yin Liao

Department of Information Management
National Chi-Nan University
Puli, Nantou, 545, TAIWAN
dyliao@ncnu.edu.tw

Abstract - This paper deals with the daily imaging scheduling problem for a low-orbit, earth observation satellite, which belongs to a class of single-machine scheduling problems. Its salient features include sequence-dependent setup effects, job-assembly characteristics, and time window constraints. Instead of looking for the global optimal solutions, we adopt a promising evolutionary approach, Tabu Search, to solve this scheduling problem. Numerical results demonstrate that the approach is effective and efficient in applications to the real problems.

Keywords: Satellite Scheduling, Tabu Search

1 Introduction

This paper deals with the daily imaging scheduling problem for a low-orbit, earth observation satellite, ROCSAT-II [9]. The daily imaging scheduling problem considers various imaging requests with different reward opportunities, changeover efforts between two consecutive imaging tasks, cloud coverage effects, and the resource availability of the spacecraft. It belongs to a class of single-machine scheduling problems with salient features of sequence-dependent setup times, job-assembly, and the constraint of operating time windows.

The ROCSAT-II daily imaging scheduling problem is NP-hard in computational complexity [5]. For problems of such high complexity, dynamic programming and exhaustive search techniques are either too time-consuming or impractical for optimal solutions. Rule-based or heuristic approaches can reduce the computation time drastically but the resultant optimality is not guaranteed. Mathematical programming approaches, such as Lagrangian relaxation [4], have the advantage of computational efficiency when the optimization problems are decomposable. In many cases, the computational times increase almost linearly with the problem size. However, a heuristic is usually needed to modify the dual solution into a feasible solution. Tabu search [1] is a meta-heuristic designed for tackling hard combinatorial optimizations problems. Contrary to random search

approaches such as simulated annealing (SA) where randomness is extensively used, Tabu search is based on intelligent searching to embrace more systematic guidance of adaptive memory and learning.

The daily imaging scheduling problem of an earth observation satellite involves dispatching a set of imaging tasks to cameras on the satellite. Verfaillie *et al.* [8] considered this problem as a valued constraint satisfaction problem and solved by a Russian doll search (RDS) algorithm. Tounsi and David [6] modified the algorithm with the idea of successive search method. Experiment results show that their algorithm improves the performance by 11% over the traditional algorithm.

Vasquez and Hao [7] formulated the daily imaging scheduling problem of SPOT 5 [10] as a generalized version of the multi-dimensional knapsack model including a large number of binary and ternary logical constraints. They proposed a Tabu search algorithm by integrating important features of a neighbourhood, dynamic Tabu tenure mechanism, techniques for constraints handling, intensification, and diversification to determine which photographs to be taken.

Lin *et al.* [2], [3] adopted the Lagrangian relaxation and subgradient optimization technique to solve the daily imaging scheduling problem of ROCSAT-II. Based on the dual solution, a greedy heuristic is developed to re-allocate imaging tasks to a feasible schedule. This greedy heuristic is quick and easy to implement. However, it could probably be trapped in a local optimum. Intelligent search techniques such as Tabu search can help escape from the local optimal trap.

In this paper, we adopt the Tabu search approach to generate a sound satellite imaging schedule within allowable computation time. Our Tabu search algorithm consists of three Tabu steps: exploration, intensification, and diversification, which use the same Tabu-search-engine. Figure 1 shows the framework of our Tabu search algorithm.

* 0-7803-8566-7/04/\$20.00 © 2004 IEEE.

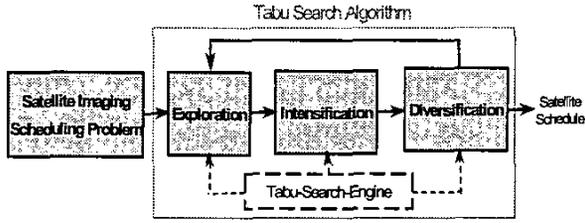


Fig 1: The Framework of Tabu Search Algorithm

Exploration is the basis of the Tabu search algorithm. It explores efficiently and extensively over constraint-related search spaces. The results include the kernel schedule, defined as the best schedule found within exploration step, and the flipping information of decision variables. These results are then used to create the search spaces for intensification and diversification steps.

Intensification focuses on the search to exploit regions of the space that the search history suggests to be promising. On the other hand, diversification undertakes to explore regions that differ from regions previously visited. In our algorithm, intensification forces the search to exploit exclusively the areas around the kernel schedule. Diversification drives the search to explore areas that are either new or not frequently visited.

The remainder of this paper is organized as follows. Section 2 presents the ROCSAT-II daily imaging scheduling problem formulation. Solution methodology is described in Section 3. Section 4 conducts the numerical experiments and demonstrates its ability in the applications to ROCSAT-II imaging scheduling. Finally, in Section 5 concluding remarks are made.

2 ROCSAT-II Imaging Scheduling

Let us define some notations before modeling the satellite imaging scheduling problem.

Notations

T	: scheduling time horizon;
t	: time period index, $t=1, \dots, T$;
J	: collection of imaging job requests;
j	: job index, $j \in J$ and $J= J $;
I_j	: collection of tasks of job j ;
I	: collection of all tasks, $I = \bigcup_{j \in J} I_j$;
i, k	: task index, $i, k \in I$ and $I= I $;
p_i	: imaging (processing) time of task i ;
C_{ki}	: unit cost of setup from processing tasks k to i ;
s_{ki}	: setup time from processing tasks k to i ;
M	: image storage capacity of Solid State Recorder;
D	: available power before the imaging operations begin;
m_i	: imaging mode of task i , $m_i \in \{\text{Panchromatic (PAN), Multi-Spectral (MS), PAN+MS}\}$;
q_i	: image size of task i ;
u_i	: power required for processing task i ;
v_{ki}	: power required for setup from task k to i ;

$[w_i^b, w_i^e]$: opportunity window of task i , w_i^b and w_i^e are the time frame available for processing task i ;

$\Gamma_i(t)$: suitability of task i at time t ;

A_j : penalty of incompleteness of job j , $A_j \geq 0$;

B_i : suitability benefit of task i , $B_i \geq 0$;

$[w_n^b, w_n^e]$: the n th cloud coverage area, $n=1, \dots, N$;

b_i : the time when task i starts its processing;

α_{it} : a step function indicating that task i is processed, where $\alpha_{it} = \begin{cases} 1, & \forall t \geq b_i; \\ 0, & \forall t < b_i; \end{cases}$

β_j : a binary variable indicating job j is complete, where $\beta_j = \begin{cases} 0, & \text{if } \alpha_{i(T-p_i+1)} = 1, \forall i \in I_j; \\ 1, & \text{otherwise;} \end{cases}$

γ_{kit} : decision variable for setup from processing tasks k to i , where $\gamma_{kit} = \begin{cases} 1, & \forall t \geq b_i - s_{ki}; \\ 0, & \text{otherwise;} \end{cases}$

Define a task to be a basic operation of image acquisition over an area of the earth. Since an imaging request may need more than one tasks, let a job be the collection of all the tasks to fulfill the request. Some assumptions are made as follows.

1. A task can belong to a job only and can only be processed at most once during the time horizon.
2. Only a task is processed or setup at a time.
3. All the imaging requests are released and given at the beginning of the scheduling time horizon.
4. There are N distinct areas with cloud coverage above them during the scheduling time horizon.

Initial Setup Constraint: Assume that the initial state of the satellite is setup to a dummy task, 0, where, $s_{0i} = 0$, $C_{0i} = 0$, and $v_{0i} = 0$, $\forall i \in I$. As there is one and only one task that can be setup from task 0, we have

$$\sum_{i=1}^I \gamma_{0iT} = 1. \quad (1)$$

Setup Constraints: An imaging operation cannot commence its processing before completing its setup. We have

$$\alpha_{it} = \sum_{\substack{k=0 \\ k \neq i}}^I \gamma_{kit(t-s_{ki})}, \quad \forall i, t. \quad (2)$$

Machine Capacity Constraints: Since there is only one camera equipped with the satellite, at any time, there is at most one task being processed or setup on the satellite, that is,

$$\sum_{i=1}^I [\alpha_{it} - \alpha_{i(t-p_i)}] + \sum_{i=1}^I \sum_{\substack{k=0 \\ k \neq i}}^I [\gamma_{kit} - \gamma_{kit(t-s_{ki})}] \leq 1, \forall t. \quad (3)$$

Storage Capacity Constraint: The images acquired are first stored on board until they can be downloaded towards

a ground station. As the total available memory on board is limited, this may impose constraints on the selection of images as well as their scheduling. The total size of images taken should be less than the available image storage capacity before imaging operations take place.

$$\sum_{i=1}^I \sum_{t=1}^T g_i (\alpha_{it} - \alpha_{i(t-p_i)}) \leq M. \quad (4)$$

Power Consumption Constraint: The total power consumption for imaging and setup operations should be less than the available power, D .

$$\sum_{i=1}^I \sum_{t=1}^T u_i [\alpha_{it} - \alpha_{i(t-p_i)}] + \sum_{i=1}^I \sum_{k=0}^I \sum_{t=1}^T v_{ki} [\gamma_{kit} - \gamma_{ki(t-s_{ki})}] \leq D. \quad (5)$$

Window of Cloud Coverage: The mission of ROCSAT-II is to acquire substantially cloud-free images. We accomplish this by employing cloud coverage prediction data sets from the weather forecast data from Central Weather Bureau (CWB). Any task that intends to take images of a cloud-covered area is considered invalid.

$$b_i \notin \{w_n^b - p_i, w_n^c\} \quad \forall i, n = 1, \dots, N. \quad (6)$$

Binary Constraints: As the variables α , β , and γ are all of either 0's or 1's, the following binary constraints should be satisfied.

$$\alpha_{it} \in \{0, 1\}, \quad \forall i \in I, t. \quad (7-1)$$

$$\beta_j \in \{0, 1\}, \quad \forall j \in J. \quad (7-2)$$

$$\gamma_{kit} \in \{0, 1\}, \quad \forall k, i \in I, t. \quad (7-3)$$

The objective of our satellite imaging scheduling problem has three folds: (i) to minimize the weighted number of incomplete jobs, (ii) to maximize the suitability benefits of imaging within the window of opportunity, and (iii) to minimize the total setup costs incurred. Mathematically, it is formulated as

$$\begin{aligned} \min_{\gamma} & \sum_{j=1}^J A_j \beta_j - \sum_{i=1}^I \sum_{t=1}^T B_i \Gamma_i(t) [\alpha_{it} - \alpha_{i(t-p_i)}] \\ \text{(P)} & + \sum_{i=1}^I \sum_{k=0}^I \sum_{t=1}^T C_{ki} [\gamma_{kit} - \gamma_{ki(t-s_{ki})}] \end{aligned}$$

subject to constraints (1) to (7).

In (P), the first term is for the weighted number of incomplete jobs, the second for preferences of placements within the window of opportunity, and the third for the costs incurred by setup operations. For tackling the job-assembly effects, the weighted number of incomplete jobs is decoupling into the weighted number of incomplete tasks. Hence, a new mathematical formulation is defined as (P'), which serves as a lower bound of (P) and is solved by our Tabu search algorithm.

$$\begin{aligned} \min_{\gamma} & \sum_{i=1}^I \left(\frac{\bar{A}_i}{\bar{\Gamma}_i} \right) [1 - \alpha_{i(T-p_i+1)}] + \sum_{i=1}^I \sum_{k=0}^I \sum_{t=1}^T C_{ki} [\gamma_{kit} - \gamma_{ki(t-s_{ki})}] \\ \text{(P')} & - \sum_{i=1}^I \sum_{t=1}^T B_i \Gamma_i(t) [\alpha_{it} - \alpha_{i(t-p_i)}] \end{aligned}$$

subject to constraints (1) to (7).

3 Tabu Search Scheduling

3.1 Unconstrained and Constrained Search Space

Define the unconstrained search space \mathcal{S} to be composed of all binary vectors of I elements, where $\mathcal{S} = \{(\alpha_{1t}, \alpha_{2t}, \dots, \alpha_{It}) \in \{0, 1\}^I\}$. Note that the size of \mathcal{S} may be huge for large number of I . Moreover, a solution has to satisfy all the constraints. Denote \mathcal{C} be the constrained search space that is composed of all binary vectors of I elements, satisfying the initial setup constraint, setup constraints, machine capacity constraints, and windows of cloud coverage, i.e.,

$$\mathcal{C} = \{s \in \mathcal{S} \mid s \text{ satisfies constraints (1) to (3), (6), and (7)}\}.$$

Vasquez and Hao [7] indicate that high quality sub-optimal solutions are located at the frontier of feasibility. Often these solutions are difficult to reach uniquely from the feasible side. A more effective way is to allow the search to oscillate around the feasibility frontier, increasing the chance to reach good solutions. Hence, we relax the constraints of storage capacity (3) and power consumption (4) to accelerate the search.

3.2 Neighborhood and Move

Define neighborhood function $N: \mathcal{C} \rightarrow (2^{\mathcal{C}} - \emptyset)$ as follows. Let $s = (\alpha_{1t}, \alpha_{2t}, \dots, \alpha_{It}) \in \mathcal{C}$. $s' = (\alpha'_{1t}, \alpha'_{2t}, \dots, \alpha'_{It})$ is a neighbour of s , i.e. $s' \in N(s)$, if and only if the following conditions are satisfied: (1) There exists one and only one task i such that $\alpha_{it} = 0$ and $\alpha'_{it} = 1$ ($1 \leq i \leq I$); and (2) $s' = (\alpha'_{1t}, \alpha'_{2t}, \dots, \alpha'_{It}) \in \mathcal{C}$.

Thus, s' can be obtained by adding an imaging task, say α_{it} , into s and at the same time removing some tasks from s to maintain the feasibility of the resultant schedule s' . Let $mv(i) = (\alpha_{it} : 0 \rightarrow 1) \cup (\alpha_{kt} : 1 \rightarrow 0, k \in I, k \neq i)$ denote such a move. The number of possible moves from a schedule s equals the number of variables in s with its value equal to 0. Let $Z = \{\alpha_{it} = 0 \mid \alpha_{it} \in s, 1 \leq i \leq I\}$. $N(s)$ has exactly $|Z|$ neighbouring schedules.

From the definition of decision variable α_{it} , the switch of its value from 0 to 1 represents not only which image but also determine the time to take the image.

3.3 Tabu List Management

A Tabu list \mathcal{T} is to prevent the search from short-term cycling. For example, $1 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow \dots$. Each time when a move $mv(i) = (\alpha_{ji} : 0 \rightarrow 1) \cup (\alpha_{ki} : 1 \rightarrow 0, k \in I, k \neq i)$ carries out, the moves $mv(j) = (\alpha_{ji} : 0 \rightarrow 1)$ and $mv(k) = (\alpha_{ki} : 0 \rightarrow 1)$ will be the classified Tabu for some following iterations (Tabu tenure) to avoid resetting the values of α_{ji} and α_{ki} from 0 to 1. The number of iterations, $iter(m)$, during which a move $mv(m)$, $m = j, k$, is the classified Tabu is dynamically defined as below [6].

$$iter(m) = a \times freq(m), a \text{ is a constant}$$

where $freq(m)$ is the number of times that α_{mi} has been flipped from 1 to 0 from the beginning of the search. Therefore, the Tabu tenure of a move depends on the flipping frequency of the decision variable, which is to penalize a move that repeats too often.

In order to implement the Tabu list, a vector Φ of I elements is used. Each element $\Phi(i)$ ($1 \leq i \leq I$) records the sum of $iter(i)$ and the current number of iterations. In this way, it is very easy to know if a move $mv(i)$ is a Tabu or not at iteration l . If $\Phi(i) > l$, $mv(i)$ is a forbidden move; otherwise, $mv(i)$ is a possible move.

3.4 Resolution of Constraint Violation

During the iterative searching in Tabu-Search-Engine, the storage capacity constraint (3) and power consumption constraint (4) are relaxed. That is, after the searching the neighborhoods of current schedule s , the storage capacity and power consumption constraints may be violated by the resultant schedule. Each time when there is any constraint violation, the scheduled task is removed to minimize the cost and a new setup relationship is then built. Continue this step until no relaxed constraint is violated.

3.5 Tabu Search Imaging Scheduling Algorithm

The stopping criterion is defined by a maximum number of iterations allowed. It is proportional to the problem size. Each Tabu step is triggered and stops automatically by the Tabu list management, whenever no more moves are admissible. The Tabu search algorithm is given below.

Tabu Search Imaging Scheduling Algorithm (TSISA)

Step 0: // Initialization
 0.1 $\mathcal{T} \leftarrow \emptyset$, ker^* , s , and $s^* \leftarrow \{0, 0, \dots, 0\}$
 // Initialize the Tabu list, ker^* , s , and s^*
 0.2 $count \leftarrow 0$ // Iteration number
while $count < count_max$
Step 1: // Exploration
 1.1 Call Tabu-Search-Engine with \mathcal{C}
 1.2 Compute I , \mathcal{D} with $freq$ and ker^*

1.2.1 $I_{=0}$ be the set of indices of ker^* that are equal to 0 and $I_{=1}$ those equal to 1.

1.2.2 $I \subseteq I_{=0}$ contains the indices of elements where flipping of these elements does not affect the elements of $I_{=1}$ even after repairing the constraint violations.

1.2.3 \mathcal{D} collects the indices of elements having a flipping frequency lower than the average.

1.3 $\mathcal{T} \leftarrow \emptyset$, $s \leftarrow ker^*$

Step 2: // Intensification

2.1 Call Tabu-Search-Engine with I

2.2 $\mathcal{T} \leftarrow \emptyset$, $s \leftarrow \{0, 0, \dots, 0\}$

Step 3: // Diversification

3.1 Call Tabu-Search-Engine with \mathcal{D}

3.2 $\mathcal{T} \leftarrow \emptyset$, $ker^* \leftarrow \{0, 0, \dots, 0\}$, $s \leftarrow$ best solution found from step 3.1

3.3 $count \leftarrow count + 1$

// end of while

Step 4: // Output s^* .

if (not $\forall \alpha_{ii} = 1$, $1 \leq i \leq I$ within s^*)

then do output \ll "the schedule is infeasible"

else

do output \ll "the schedule is feasible"

// end of if

Tabu-Search-Engine

Step 0: // Assumption

0.1 Let s and s^* be respectively the current and the best schedules.

0.2 $ker \leftarrow \emptyset$, $freq \leftarrow \{0, 0, \dots, 0\}$

0.3 Let $\mathcal{M}(s)$ be the set of candidate moves from s ($\mathcal{M}(s)$ induces a particular search space \mathcal{C} , I , or \mathcal{D}).

Step 1: // Searching

while there are possible moves exist in $\mathcal{M}(s)$

1.1 Compute the best move $mv(i)$ (break ties randomly).
 // $mv(i) \leftarrow \arg \min_{s \in \mathcal{N}(s)} f(s'), f(\bullet)$: primal function.

1.2 $s \leftarrow s + mv(i)$, $freq(i)++$, $\mathcal{T} \leftarrow \mathcal{T} \cup \{mv(i)\}$,
 $ker \leftarrow ker \cup s$

1.3 // Resolve the violations on storage capacity constraint and power consumption constraint (for Exploration and Intensification).

while there is any relaxed constraint violation

1.3.1 Remove task i^* from s , where i^* is the index of
 $\alpha_{i^*} \in s$ and $s \leftarrow \arg \max_s \{f(s) - f(s \setminus \{\alpha_{i^*}\})\}$.

1.3.2 Adjust setups among the remaining tasks in s

// end of while

1.3.3 **if** $f(s) < f(s^*)$ **then** $s^* \leftarrow s$

// end of while

4 Numerical Experiments

Numerical experimentation is conducted to assess the performance of the proposed satellite daily imaging scheduling algorithm. Features of job assembly, setup effects, cloud coverage, and opportunity windows are considered in the test cases. The algorithm is first tested with a projected ROCSAT-II daily imaging scenario to demonstrate its applicability to the realistic problem.

Algorithmic properties of both computational efficiency and optimality are further explored with respect to job-assembly, setup effects, and time window constraints, respectively. In order to demonstrate the efficiency and effectiveness of our approach, the scheduling algorithm based on Lagrangian relaxation approach [2], [3] is adopted as the comparative algorithm.

We implement both the Tabu search and Lagrangian relaxation algorithms in C language. All the experiments are conducted in a PC with dual INTEL Xeon-2.4GHz CPUs with 1 GB memory.

4.1 ROCSAT-II Daily Imaging Scheduling

Consider a test scenario of the daily imaging scheduling problem of ROCSAT-II. As shown in Fig. 2, there are five imaging jobs, Jobs 1-5, which are composed of nine tasks. These five imaging jobs consider the conditions of wide-spread tasks (Job 1), large-area tasks (Job 2), separate-but-in-a-same-strip tasks (Job 3), cross-stripped tasks (Job 4), and consecutive tasks (Job 5), respectively. The time window of each task is determined by the geographical limitation of the task. The setup time between any two tasks is proportional to their geographical distance, and the setup cost is twice the value of the corresponding setup time. The parameters and coefficients of the cost functions are listed in TABLE 1. The scheduling time horizon is of 100 time periods. Two cloud coverage areas are assumed with totally 4 time periods of invalid imaging operations.

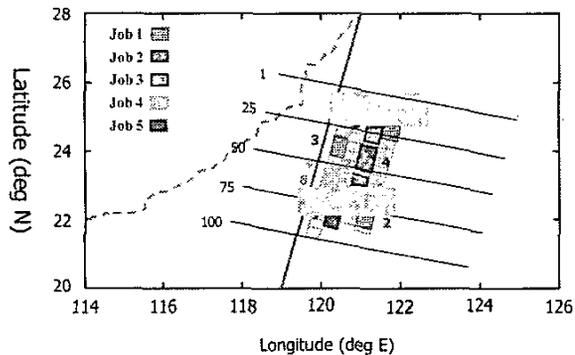


Fig. 2. A Projected Case (Source: NSPO)

Define imaging loading as the ratio of total processing times over the scheduling horizon. Three test cases of light, heavy and overloaded imaging loads are designed. The required imaging time is 67 time periods for the light-loaded case, 80 for the heavy-loaded case, and 93 for the overloaded one.

TABLE 1. 5 JOBS, 9 TASKS IN 100 TIME PERIODS

Job No.	Task No.	A _j	B _i	Window	Imaging Time (s)			Setup Time (s) ^{**}								
					Light	Heavy	Over	1	2	3	4	5	6	7	8	9
1	1	33	15	[1, 36]	7	8	9	8	4	3	2	1	2	4	5	3
1	2	39	59	[60, 101]	7	8	9	4	0	5	2	4	2	3	4	1
1	3	33	58	[71, 94]	7	8	9	3	8	0	1	2	3	3	1	7
1	4	60	38	[50, 60]	27	12	16	3	3	1	6	1	3	2	8	1
1	5	49	32	[18, 43]	8	6	8	1	4	2	1	0	2	1	4	4
3	6	43	35	[46, 85]	6	8	8	3	2	2	1	2	8	0	1	3
4	7	43	35	[46, 85]	6	8	8	4	3	2	2	3	8	0	8	1
4	8	43	35	[46, 85]	6	8	8	3	4	1	3	4	1	0	8	2
5	9	308	43	[87, 97]	13	12	14	3	1	3	3	4	2	1	3	8

^{**}: setup cost = 2.0 × s_{ki}.

Numerical results of these three test cases are summarized in TABLE 2. The proposed Tabu search approach gets sound schedules for light and heavy loading cases, respectively. No feasible schedule is obtained for the overloaded case. TABLE 2 also shows the comparison results of both our proposed approach and the Lagrangian relaxation approach. Note that the proposed Tabu search approach is superior to Lagrangian relaxation approach in optimality but with little increase in computation time.

TABLE 2. Comparison Results

Problem	Tabu Search				Lagrangian Relaxation			
	Scheduled		Primal Cost	CPU Time (Seconds)	Scheduled		Primal Cost	CPU Time (Seconds)
	Setup	Imaging			Setup	Imaging		
Toy Example	2	4	-58.5	0.0014	2	4	-58.5	0.0008
Light Load	9	67	-294.0	1.1670	7	67	-306.0	0.4465
Heavy Load	7	80	-306.0	0.8655	7	80	-304.8	0.4935
Over Load	6	84	-241.056*	0.9125	9	77	-161.0*	0.5345

*: Infeasible

4.2 Algorithmic Features

In order to further explore the algorithmic features of our satellite daily imaging scheduling algorithm, 19 test cases are designed and simulated. All the test data are based on a realistic problem (case No. 5 in TABLE 3) as their baseline data set. The baseline problem considers 25 imaging jobs ($J = 25$) with 100 tasks ($I = 100$) to be scheduled within a time frame of 600 seconds ($T = 600$) when the satellite traverses the vicinity of the target area. The setup time and cost between any two consecutive tasks are randomly generated with mean setup time and variance to be one second, respectively. For each imaging task, its opportunity window, suitability function, and imaging mode are all randomly generated. The imaging size and power consumption factors of each task are assumed to be identical. They are 1 for PAN, 1.5 for MS, and 2 for PAN+MS. The power required to setup from task k to task i , is set to be $v_{ki} = 1.0 \times s_{ki}, \forall k, i$. For all cases, their corresponding coefficients in the objective function are set to $A_j = 150, \forall j, B_i = 15, \forall i$, and $C_{ki} = 2.0 \times s_{ki}, \forall k, i$. Each test case is simulated with 10 scenarios randomly. The loading of each test case is set to 66.7%. Simulation results are summarized in TABLE 3.

The results of Group I indicate that there are no significant differences in optimality for different numbers of jobs. Also, the number of jobs has no impacts on the computational time. We therefore conclude that number of jobs has no direct impacts on both the optimality and the computational time. In order to study the setup effects on the satellite scheduling algorithm, five test cases are designed in Group II. Results indicate that the costs become worse and the computational times become larger as the increase of setup times. The effects due to different lengths of time windows are designed and studied in Group III. Note that the average costs become worse when the lengths of time windows decrease. However, the impact of computational time is not significant.

TABLE 3. SIMULATION OF 19 TEST CASES

Case	Group	Dimension Factor			Loads		Effects		Performance Index (Average)			
		J	I	T	Imaging (p)	Setup	Time Window	Tabu Search CPU	Cost	Lagrangian CPU	Cost	
1	I	1	100	600	4	50%	25 _{sp}	261	-1417.8	251.1	-1498.7	
2		5	100	600	4	50%	25 _{sp}	311.2	-1411.1	248.6	-1353.2	
3		10	100	600	4	50%	25 _{sp}	311.7	-1401	250.5	-1369.2	
4		20	100	600	4	50%	25 _{sp}	325.8	-1413.2	230.6	-1389.2	
5		25	100	600	4	50%	25 _{sp}	337.1	-1401.5	249.5	-1389.6	
6		50	100	600	4	50%	25 _{sp}	296.5	-1401.8	251.5	-1406.6	
7		100	100	600	4	50%	25 _{sp}	385.2	-1395.2	254.1	-1373.2	
8	II	25	100	600	4	0%	25 _{sp}	284.5	-1499.7	238.6	-1497.8	
9		25	100	600	4	25%	25 _{sp}	315.9	-1476.7	242.9	-1483.3	
10		25	100	600	4	75%	25 _{sp}	323.8	-1354	255.1	-1292.7	
11	III	25	100	600	4	100%	25 _{sp}	380.5	-1077.5	260.9	-1062.9	
12		25	100	600	4	50%	1 _{sp}	221.4	-786.7	159.6	-703.5	
13		25	100	600	4	50%	5 _{sp}	343.2	-1693.8	224	-1040.9	
14		25	100	600	4	50%	10 _{sp}	361.8	-1226.6	229.3	-1207.5	
15		25	100	600	4	50%	50 _{sp}	337.9	-1457.5	264.8	-1462.7	
16		25	100	600	4	50%	75 _{sp}	358.3	-1488.2	273.4	-1448.2	
17		25	100	600	4	50%	100 _{sp}	355.5	-1493.1	285.8	-1425.7	
18	25	100	600	4	50%	125 _{sp}	322.5	-1499.8	294.9	-1447.4		
19	25	100	600	4	50%	150 _{sp}	250.6	-1500	234.3	-1421.2		

CPU: CPU Time (in seconds), Cost: Primal Cost

In summary, most of the CPU times spent in the regular cases (in the same size of the baseline problem) are less than 385 seconds. The costs in most of the test cases are less than those of the Lagrangian relaxation approach. Therefore, we conclude that this daily imaging scheduling algorithm is quite efficient for realistic applications to provide a sound imaging schedule to the earth observation satellite.

5 Conclusions

This paper presents the development of a daily imaging scheduling system of an earth observation satellite, ROCSAT-II. The daily imaging scheduling problem of satellite belongs to single machine scheduling problems with salient features of sequence-dependent setup, job-assembly, and constrained operating time windows. It is formulated as an integer programming problem, which is NP-hard in computational complexity. We have developed an effective Tabu search algorithm to solve this problem. This Tabu search algorithm combines some ideas to make the searching effective and efficient. Those ideas include constrained neighborhood searching, greedy-based flipping process, and frequency-based dynamic Tabu tenure. Intensification and diversification

steps are used to avoid of being trapped in local optimum. Numerical results on three test scenarios of the daily imaging scheduling problem of ROCSAT-II and 19 realistic problems have shown the effectiveness of this algorithm. Comparison results also show our algorithm is superior to the Lagrangian relaxation approach in optimality but with little increase of computational time.

Future research may extend the algorithm to include realistic issues such as seasonal refreshment and coordination between multiple satellites. On the other hand, the developed algorithm deals with the scheduling problem assuming no machine failure and exclusions of imaging with cloud coverage. Extensions to this research may handle these stochastic issues.

References

- [1] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.
- [2] W.-C. Lin, D.-Y. Liao, C.-Y. Liu, and Y.-Y. Lee, "Daily Imaging Scheduling of An Earth Observation Satellite," *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 2, pp. 1886-1891, 2003.
- [3] W.-C. Lin, D.-Y. Liao, C.-Y. Liu, and Y.-Y. Lee, "Daily Imaging Scheduling of An Earth Observation Satellite," to appear in *IEEE Transactions on Systems, Man, and Cybernetics*, 2004.
- [4] P. B. Luh and D. J. Hootom, "Scheduling of Manufacturing Systems Using the Lagrangian Relaxation Technique," *IEEE Trans. on Automatic Control*, Vol. 38, No. 7, pp. 1066-1079, 1993.
- [5] M. Pinedo, *Scheduling Theory, Algorithms, and Systems-2nd Edition*, Prentice Hall 2002.
- [6] M. Tounsi and P. David, "Successive Search Method for Valued Constraint Satisfaction and Optimization Problems," *Proceedings of the 13th International Conference on Tools with Artificial Intelligence*, pp. 341-347, 7-9 Nov. 2001.
- [7] M. Vasquez and J. K. Hao, "A Logic-Constrained Knapsack Formulation and A Tabu Algorithm for the Daily Photograph Scheduling of An Earth Observation Satellite," *Computational Optimization and Applications* 20, No. 2, pp. 137-157, 2001.
- [8] G. Verfaillie, M. Lemaitre, and T. Schiex, "Russian Doll Search for Solving Constraint Optimization Problems," In *AAAI-96*, Portland, Oregon, USA, 1996.
- [9] NSPO: <http://www.nspo.gov.tw/>.
- [10] Spot Image Spot 5: <http://www.spotimage.fr/home/system/future/spot5/welcome.htm>.