

Article

Instance and Data Generation for the Offline Nanosatellite Task Scheduling Problem

Cezar Antônio Rigo ¹, Edemar Morsch Filho ², Laio Oriel Seman ^{3,4,*}, Luís Loures ⁵
and Valderi Reis Quietinho Leithardt ⁵

¹ Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianópolis 88040-900, Brazil

² Department of Aeronautical Engineering, São Paulo State University (UNESP), São João da Boa Vista 13876-750, Brazil

³ Graduate Program in Applied Computer Science, University of Vale do Itajaí (UNIVALI), Itajaí 88302-901, Brazil

⁴ Industrial and Systems Engineering Graduate Program, Pontifical Catholic University of Parana, Curitiba 80215-901, Brazil

⁵ VALORIZA—Research Center for Endogenous Resources Valorization, Instituto Politécnico de Portalegre, 7300-555 Portalegre, Portugal; lcloures@ipportalegre.pt (L.L.); valderi@ipportalegre.pt (V.R.Q.L.)

* Correspondence: laio@univali.br

Abstract: This paper discusses several cases of the Offline Nanosatellite Task Scheduling (ONTS) optimization problem, which seeks to schedule the start and finish timings of payloads on a nanosatellite. Modeled after the FloripaSat-I mission, a nanosatellite, the examples were built expressly to test the performance of various solutions to the ONTS problem. Realistic input data for power harvesting calculations were used to generate the instances, and an instance creation procedure was employed to increase the instances' difficulty. The instances are made accessible to the public to facilitate a fair comparison of various solutions and to aid in establishing a baseline for the ONTS problem. Additionally, the study discusses the various orbit types and their effects on energy harvesting and mission performance.

Keywords: instance generation; nanosatellite; QoS; ONTS



Citation: Rigo, C.A.; Morsch Filho, E.; Seman, L.O.; Loures, L.; Leithardt, V.R.Q. Instance and Data Generation for the Offline Nanosatellite Task Scheduling Problem. *Data* **2023**, *8*, 62. <https://doi.org/10.3390/data8030062>

Academic Editors: Yu Tao, Siting Xiong and Rui Song

Received: 2 February 2023

Revised: 3 March 2023

Accepted: 5 March 2023

Published: 21 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

This paper offers a collection of instances of the ONTS (Offline Nanosatellite Task Scheduling) optimization problem, which aims to schedule the activation and deactivation times of payloads on a nanosatellite. Nanosatellites are revolutionizing the space industry by providing a cost-effective and versatile platform for various space applications, from remote sensing and earth observation to communication and technology demonstrations [1–3]. They are a vigorous area of research, and more recent methodologies address mission planning optimization, which requires real and repeatable input data for scientific exploration and benchmarks [4–8].

In order to improve power management, guarantee Quality of Service (QoS), and gather more data throughout the satellite mission lifetime, these studies have investigated several strategies to improve the scheduling of jobs for nanosatellites. The solutions discussed aim at effective resource management while taking into account variables such as available power, task characteristics, and satellite resources. In [5], a battery model and fuzzy constraints have been added to a mathematical integer programming approach to maximize the number of jobs that can be completed in space while promoting energy-efficient management and extending battery life. For large cases of the problem, a branch-and-price method using dynamic programming has been presented as part of a solution technique that uses the Dantzig–Wolfe decomposition [4]. In [8], time synchronization, mission quality of service, and shared resource management have been included in a comparison of

continuous-time versus discrete-time approaches. These methods provide decision makers more freedom and alternatives when it comes to planning the work that needs to be carried out by nanosatellites.

The instances used in these studies were created expressly to benchmark the performance of various solution methodologies to the ONTS problem. Nevertheless, the methodology for creating such instances was never published before, which is the intent of this paper. The instances are modeled after the FloripaSat-I mission, a CubeSat that has been validated in orbit [9]. This spacecraft format has recently gained significant market share given its cost-effective and simpler deployment [10]. Based on the orbital parameters of the FloripaSat-I mission, realistic input data for power harvesting calculations were used to construct the examples presented in this paper. Furthermore, by using the generation procedure based on random data described here, the authors could scale the difficulty and create instances on demand.

It is necessary to emphasize that sharing examples for a particular subject is essential for scientific study repeatability. Other researchers can use the same set of instances to evaluate their approaches and compare their results with those reported in the publication if the instances are made publicly available. This enables a fair comparison of various solutions and contributes to establishing a benchmark for the ONTS problem. In addition, having a publicly accessible set of cases for a particular subject might help to improve the field's research by offering a common starting point for new investigations.

In order to plan a nanosatellite mission effectively and ensure that the spacecraft can complete its mission objectives, it is crucial to estimate its power budget. The energy input can be computed on a minute-by-minute basis according to orbit and attitude parameters and account for variables such as EPS efficiency, which can impact the amount of available energy for operations. These data are essential for the design phase because they enable engineers to make informed judgments on the amount and type of payloads that can be included on the satellite.

For instance, if the power calculation reveals that the satellite will not have enough energy to perform a specific duty at the necessary frequency, the engineer may need to lower the number of payloads or modify the design to improve energy efficiency. Knowing the estimated power is also essential for scheduling satellite tasks. Offline scheduling algorithms can use this information to find the optimal times to activate and deactivate payloads to maximize available energy usage. This can ensure that the satellite can complete all its responsibilities within the allotted time.

This paper's objectives are to provide a methodology as well as ready-to-use problems of the Offline Nanosatellite Task Scheduling (ONTS) optimization problem. With the use of actual input data for power harvesting calculations based on the FloripaSat-I mission, the study seeks to provide realistic means to compare different solution techniques for the ONTS issue. The study's examples and methods are created to help researchers assess their approaches and contrast their findings with those in the published works. By providing a standard place to begin future investigations and creating a baseline for the ONTS problem, this study's ultimate purpose is to advance the area of research.

2. Problem Formulation

The Offline Nanosatellite Task Scheduling (ONTS) problem can be fully captured by modeling the tasks, or payloads, required behavior, such as to guarantee mission Quality of Service (QoS) and the power generation and management systems. These two main issues are presented in this section.

2.1. Task Modeling

According to the payloads they represent, the ONTS problem's tasks exhibit particular behaviors. Four major requirements must be met in order to schedule these jobs. To begin, we assume that tasks are non-preemptive. This means that once a work has begun, it must

be finished satisfactorily within a certain time frame. This is the scheduling problem's initial requirement.

Second, jobs are normally performed within a defined time frame. This means that a minimum and maximum duration between task activations must be specified. This is the scheduling problem's second requirement.

Third, tasks must start a minimum number of times during an orbit to complete mission criteria. An upper constraint can restrict this number if a job must only be executed once. This is the scheduling problem's third requirement.

Finally, tasks can optimize their optimal start and completion timings to ensure the highest potential mission value. This is the scheduling problem's fourth requirement. Another key feature for ensuring service quality is scheduling based on each task priority u_j . Table 1 presents an overview of such QoS parameters.

Table 1. Tasks' QoS requirements.

Notation	Definition
J	Number of tasks.
T	Number of time units.
u_j	Priority of task j .
t_j^{\min} / t_j^{\max}	Minimum and maximum running time required by task j .
y_j^{\min} / y_j^{\max}	Minimum and maximum startups required by task j .
p_j^{\min} / p_j^{\max}	Minimum and maximum period required by task j .
w_j^{\min} / w_j^{\max}	Start and finish time of a task j execution window.

These requirements are critical for modeling operations such as environmental data collecting, image capture, or other high-level activity carried out by dedicated hardware, namely sensors, transponders, payloads, cameras, and similar, which substantially influence the satellite's energy consumption.

Constellation Requisites

Mission planning, or task scheduling, for a group of nanosatellites requires careful consideration of various factors. For instance, each satellite can fulfill constellation duties; hence, it is important to specify a simultaneity factor for each task. This is accomplished through constraints that limit the minimum and maximum number of subsystems required to perform a task at any given time. Additionally, lower and upper bounds for the number of startups must be specified for constellation task assignments. This is identical to the previously mentioned limitations, except now that all satellites in the constellation are examined.

Network behavior is another key factor to examine. Some jobs may benefit from running synchronously over the entire constellation simultaneously, and this can be enforced via a simultaneous start and stop. This can be accomplished by imposing constraints requiring all satellites to perform the same task simultaneously.

To plan a constellation mission with redundant task execution, one can add a binary flag made equal 1 whenever two jobs should run simultaneously in distinct spacecrafts and 0 otherwise. The purpose of this is to ensure mission reliability by redundancy. Additionally, the minimum and maximum global number of startups for each task must be optimized.

Therefore, the ONTS problem in a nanosatellite constellation involves a balance between coordination among the satellites, meeting mission requirements and power constraints while ensuring a high level of reliability and redundancy.

2.2. Power Generation and Management

Beyond running requirements, the tasks have a power impact that must be considered, requiring energy to be completed. As a result, given accessible power resources, the aggregate of power drained by all jobs at a given time t must not be greater than the available

quantity. Controlling power in a spacecraft requires modeling orbit dynamics to obtain the energy from solar panels at every instant and a battery to act as a power reserve. This entails several parameters that represent a battery, shown in Table 2.

Table 2. Power management parameters.

Notation	Definition
V_b	Battery voltage, in Volts.
Q	Battery capacity, in Ampere-hour.
ρ	Battery level lower bound.
e_{cd}	Battery charge/discharge efficiency.
γ	Upper bound for battery discharge current, in Ampere.
r_t	Energy harvested and available for tasks at time t .
q_j	Power drained running task j .

It is important to note that γ is an upper bound for charge and discharge battery currents (C and E rates, respectively), and restricting such rates has been demonstrated to protect battery longevity [11]. Furthermore, ρ limits the battery depth-of-discharge, a critical feature for mission margin of safety and battery lifespan expansion [12].

By contrast, photovoltaic cells are, by far, the most applied technology in satellite missions for power generation. Therefore, the exposition and projection of solar panels towards the sun are key aspects to be modeled. The orbital and attitude models that will be discussed in the following sections are used to estimate the power generation P (W) on a surface k of the nanosatellite, according to:

$$P_k = \eta_k G A_k F_k \xi \quad (1)$$

The total solar irradiance reaching a surface k is computed using the solar flux ($G = 1367 \text{ W/m}^2$ [13]), the solar cell surface's area (A_k (m^2)), the solar cell surface's view factor towards the sun (F_k (-)), and the occurrence of the earth's shadow (ξ (-)) [14]. However, only a fraction of this incoming radiation is actually useful for the photovoltaic phenomenon, and an extra parameter η is necessary to account for the efficiency of the cell [15].

2.2.1. Orbit Model

One major aspect of every satellite mission of any type and size is the orbital parameters. The orbit governs the nanosatellite's position, an essential parameter to estimate the irradiance field and, consequently, the power generation through photovoltaic cells. When the interest is for short periods, e.g., around one day or less, the ideal Keplerian orbit model provides relevant results for orbits higher than 300 km. This model also usually generates reasonable values for longer periods if the apogee exceeds 600 km. However, the Keplerian model cannot be adopted for more precise results and to extend its range of applications. The Keplerian orbit assumes that the only existing force comes from the gravity acceleration between two ideal bodies, i.e., the earth and the nanosatellite. In fact, a precise orbit model should include perturbations, such as the atmospheric drag, the oblateness of the earth, and solar pressure, but all of them introduce further complexities [16].

To simulate the position of a satellite in orbit, the user must always inform six orbital elements and may use the following equation:

$$\mathbf{r}_X = \mathbf{Q} \mathbf{r}_{\bar{x}} \quad (2)$$

where \mathbf{Q} is the direct cosine matrix to transform the vector from the perifocal frame to the geocentric equatorial frame, as established by the Euler angle sequence [16]:

$$\mathbf{Q} = \begin{bmatrix} -\sin \Omega \cos i \sin \omega + \cos \Omega \cos \omega & -\sin \Omega \cos i \cos \omega - \cos \Omega \sin \omega & \sin \Omega \sin i \\ \cos \Omega \cos i \sin \omega + \sin \Omega \cos \omega & \cos \Omega \cos i \cos \omega - \sin \Omega \sin \omega & -\cos \Omega \sin i \\ \sin i \sin \omega & \sin i \cos \omega & \cos i \end{bmatrix} \quad (3)$$

Three orbital elements required are the ascending node Ω ($^\circ$), the orbit inclination i ($^\circ$), and the argument of perigee ω ($^\circ$). The remaining vector $\{\mathbf{r}\}_{\bar{x}}$ is the position of the nanosatellite in the perifocal frame of reference, calculated by:

$$\mathbf{r}_{\bar{x}} = \frac{h^2}{\mu(1 + e \cos \theta)} \begin{Bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{Bmatrix} \quad (4)$$

where h is the specific relative angular momentum (km^2/s), θ is the true anomaly ($^\circ$), and e is the eccentricity of the orbit (-). These and the previous parameters complete the six orbital elements. The term μ is the gravitational parameter ($389,600 \text{ km}^3/\text{s}^2$ for geocentric orbits).

By definition, $\theta = 0^\circ$ at the perigee (closest position to the earth), while $\theta = 180^\circ$ at the apogee (furthest position to the earth), and by making $0^\circ \leq \theta \leq 360^\circ$ a complete orbit is covered. Another option is to provide a time step and increase it until the orbit period (Period) is reached. For each time there is a respective value for θ , calculated as [16]:

$$M = 2\pi \frac{t}{\text{Period}} \quad (5)$$

$$E = M + e \sin(E) \quad (6)$$

$$\theta = 2 \tan^{-1} \left(\sqrt{\frac{1+e}{1-e}} \tan \frac{E}{2} \right) + \theta_0 \quad (7)$$

where M is the mean anomaly (rad), t is time (s), E is the eccentric anomaly, and θ_0 is the initial true anomaly.

To estimate the nanosatellite's passage under the earth's shadow, it is crucial to determine its position and the position of the sun (\mathbf{r}_{sun}). Using Equation (8) [16], we can test whether the nanosatellite is in the shadow ($\xi = 0$) or not ($\xi = 1$) based on its position and the sun's location:

$$\xi = \begin{cases} 0, & \cos^{-1} \left(\frac{R_E}{|\mathbf{r}_X|} \right) + \frac{\pi}{2} \leq \cos^{-1} \left(\frac{\mathbf{r}_X \cdot \mathbf{r}_{\text{sun}}}{|\mathbf{r}_X| |\mathbf{r}_{\text{sun}}|} \right) \\ 1, & \text{otherwise} \end{cases} \quad (8)$$

where R_E is a constant value for the radius of the earth (6378 km). The estimation of the sun's position only requires information about the date (day, month, and year) [16], as will be shown later.

2.2.2. Attitude

From the earth's orbit point of view, the solar flux can be regarded as parallel; nevertheless, the kinematics of the nanosatellite can be rather complex and for prediction of energy harvesting, it is necessary to compute the orientations of the nanosatellite toward the sun along the orbit.

Attitude refers to the pointing direction of the spin axis in satellites [17]. Even if the sun shines on the spacecraft, the absorption of solar radiation is a function of the attitude. Therefore, nanosatellites on identical orbits may have surfaces under different scenarios of solar flux as a consequence of their attitudes, e.g., some arbitrary spin, RAM (the spin axis is aligned with the velocity vector), sun-fixed (one or more sides constantly towards the sun), nadir (one or more sides constantly towards the center of the earth) or magnet (here

referring to the state obtained by passive attitude control where the nanosatellite follows the earth's magnetic field). Figure 1 illustrates these attitudes.

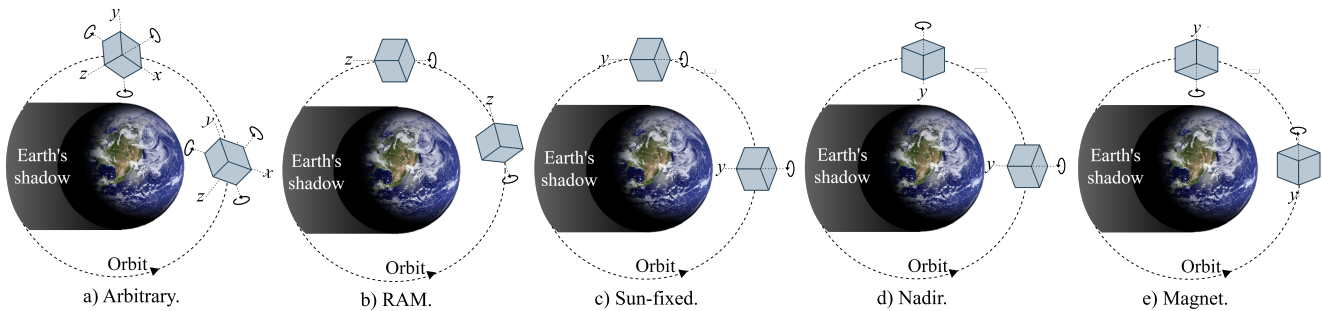


Figure 1. Examples of nanosatellites' attitude.

For any attitude and any type of satellite, the projection of a surface k of the nanosatellite towards the sun, or in other words, its view factor, may be summarized by the parameter F_k . Its value is between 0 (normal vector of surface k does not “see” the sun) and 1 (normal vector of surface k perfectly aligns with solar rays). Any other orientation will result in intermediate values (ignoring possible self-shadowing among the nanosatellite's parts).

To simulate the attitude, the initial nanosatellite's orientation and mass distribution must be known, and the forces and torques acting on it are integrated. To simplify this process, a pseudo attitude response may be obtained by specifying rotations for the satellite's axis instead of solving the differential equations that govern it. This method is adopted here so a rotation matrix \mathbf{B} and angles are informed by the user to obtain the spin of the nanosatellite.

To exemplify this process, the attitude of the nadir will be further described here. Assuming a cubic shape, as those from CubeSat 1U, the vector of initial orientation of each side k may be assumed as:

$$\mathbf{n}_{k=X+} = \{+1 \ 0 \ 0\}^T \quad (9)$$

$$\mathbf{n}_{k=X-} = \{-1 \ 0 \ 0\}^T \quad (10)$$

$$\mathbf{n}_{k=Y+} = \{0 \ +1 \ 0\}^T \quad (11)$$

$$\mathbf{n}_{k=Y-} = \{0 \ -1 \ 0\}^T \quad (12)$$

$$\mathbf{n}_{k=Z+} = \{0 \ 0 \ +1\}^T \quad (13)$$

$$\mathbf{n}_{k=Z-} = \{0 \ 0 \ -1\}^T \quad (14)$$

These directions will change with time as the nanosatellite completes the orbit so that one side always faces the earth (see Figure 1d). Therefore, the orientation $\mathbf{N}_{\bar{x}_k}$ in each side k , on the perifocal frame of reference, can be calculated by:

$$\mathbf{N}_{\bar{x}_k} = \mathbf{B} \mathbf{n}_k \quad (15)$$

where, for this specific case, the matrix of rotation is:

$$\mathbf{B} = \begin{bmatrix} \cos(\Omega) & \sin(\Omega) & 0 \\ -\sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta^*) & \sin(\theta^*) \\ 0 & -\sin(\theta^*) & \cos(\theta^*) \end{bmatrix} \quad (16)$$

The parameter θ^* is a residual spin to keep the satellite rotating but still always facing the earth. For different scenarios of attitude, the user has to combine other rotations of matrices.

Once the directions of the sides along the orbit are known, they must be translated from the perifocal frame of reference to the geocentric equatorial frame of reference [16]. Therefore:

$$\mathbf{N}_{X_k} = \mathbf{Q}\mathbf{N}_{\tilde{x}_k} \quad (17)$$

Finally, the projection of any side k results from the dot product between $\{\mathbf{N}_X\}_k$ and the solar position, given by:

$$F_k = \begin{cases} \mathbf{N}_{X_k} \cdot \frac{\mathbf{r}_{\text{sun}}}{|\mathbf{r}_{\text{sun}}|}, & \mathbf{N}_{X_k} \cdot \frac{\mathbf{r}_{\text{sun}}}{|\mathbf{r}_{\text{sun}}|} \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

3. Instance Generation

Considering all aspects presented in the problem formulation section, we now present the exact procedures to generate a realistic instance of the ONTS problem, from power calculation to tasks scheduling data.

3.1. General Procedure for Power Calculation

The basic process to determine the power input for the instances is presented in Algorithm 1. The inputs for sun irradiance (G), solar cell efficiency (η), and solar cell area (A), as well as a timestep, must first be entered into the code. The algorithm specifies a few conversion variables in the sequence that will be used later in the code. Algorithm 1 then calls Algorithm 2 (“inputOrbit”), which is used to convert Orbital Inputs (OI) provided in units of angle and to derive preliminary orbital parameters that will later be used. A Newton–Raphson-based subfunction called “kepler_E” (Algorithm 3) is included in this algorithm to compute the eccentric anomaly (E). It is important to note that the parameter ϵ , which serves as a stopping point for convergence in Algorithm 3, is a small positive integer. Algorithm 4 (solar_position) is executed following Algorithm 2 to determine the sun’s position (\mathbf{r}_{sun}) for a particular day [16]. When Algorithms 2 and 4 are complete, the loop in Algorithm 1 starts, and each iteration updates the time, the satellite’s position (\mathbf{r}_X), the eclipse status (ζ), the attitude (\mathbf{N}_X), the view factor (F), and the power generation (P) for each side of the CubeSat. When a full orbit’s period is reached, the loop comes to an end.

Five different results for power generation are illustrated in Figure 2. Except for the attitude, they have the following common parameters: $\Omega = 0^\circ$, $\omega = 0^\circ$, $i = 90^\circ$, $e = 0$, $M = 0^\circ$, and $M_m = 15.2198$ rev/day (equivalent to the altitude of 500 km), valid for 21 March 2023.

Algorithm 1: General power calculation procedure

```

Input :Solar irradiance  $G$ , solar cell efficiency  $\eta$ , solar cell area  $A$ , time step
Output:Equation (1)
/* Initialization */
1  $i \leftarrow 0$ ;
2  $t_0 \leftarrow 0$ ;
/* Conversion factors */
3 hours  $\leftarrow 3600$ ;
4 days  $\leftarrow 24 \times$  hours;
5 rad  $\leftarrow \frac{\pi}{180}$ ;
/* Constants */
6  $\mu \leftarrow 398600$ ;
7  $R_E \leftarrow 6378$ ;
/* Read Orbit Input (OI) and further arguments */
8  $[h, e, \Omega, inc, \omega, \theta_0, year, month, day, Period] \leftarrow \text{inputOrbit}(\mu, \text{rad}, \text{OI})$ ;
/* Compute the position of the sun on a given day */
9  $\mathbf{r}_{\text{sun}} \leftarrow \text{solar\_position}(\text{year}, \text{month}, \text{day})$ ;
10 while  $t_0 \leq \text{Period}$  do
11    $i \leftarrow i + 1$ ;
12   time $i$   $\leftarrow (i - 1) \times \text{timestep}$ ;
13   /* transformation from time [s] to true anomaly [rad] */
14    $\theta \leftarrow \text{Equations (5)–(7)}$ ;
15   /* position of the satellite */
16    $\mathbf{r}_X \leftarrow \text{Equations (2)–(4)}$ ;
17   /* eclipse of the earth */
18    $\xi \leftarrow \text{Equation (8)}$ ;
19   /* attitude of the satellite */
20    $\mathbf{N}_X \leftarrow \text{Equations (9)–(17)}$ ;
21   /* view factor */
22    $F \leftarrow \text{Equation (18)}$ ;
23   /* power generation */
24    $P_i \leftarrow \text{Equation (1)}$ ;
25    $t_0 \leftarrow \text{time}_i$ ;
26 end

```

Algorithm 2: Function—inputOrbit

```

Input : $\mu, \text{rad}, \text{OI}$ 
Output: $h, e, \Omega, inc, \omega, \theta_0, T$ 
1  $\Omega \leftarrow \text{OI}(1) \times \text{rad}$ ;
2  $inc \leftarrow \text{OI}(2) \times \text{rad}$ ;
3  $\omega \leftarrow \text{OI}(3) \times \text{rad}$ ;
4  $e \leftarrow \text{OI}(4) \times \text{rad}$ ;
5  $M \leftarrow \text{OI}(5) \times \text{rad}$ ;
6  $E \leftarrow \text{kepler\_E}(e, M)$ 
7  $\theta_0 \leftarrow 2 \times \tan^{-1}(\sqrt{\frac{1+e}{1-e}} \tan \frac{E}{2})$ 
8  $M_m \leftarrow \text{OI}(6) \times 2\pi / (24 \times 3600)$ 
9  $a \leftarrow (\mu^{1/3}) / M_m^{2/3}$ 
10  $h \leftarrow \sqrt{a\mu(1-e^2)}$ 
11  $\text{Period} \leftarrow 2\pi(a^{1.5}) / \sqrt{\mu}$ 

```

Algorithm 3: Function—kepler_E

```

Input : $e, M$ 
Output: $E$ 
1  $\epsilon \leftarrow 10^{-8}$ 
2  $i \leftarrow 0$ 
3  $E_i \leftarrow M$ 
4 while  $|E_{i+1} - E_i| > \epsilon$  do
5    $E_{i+1} \leftarrow E_i + \frac{M + e \sin(E_i) - E_i}{1 - e \cos(E_i)}$ 
6    $i \leftarrow i + 1$ 
7 return  $E_i$ 

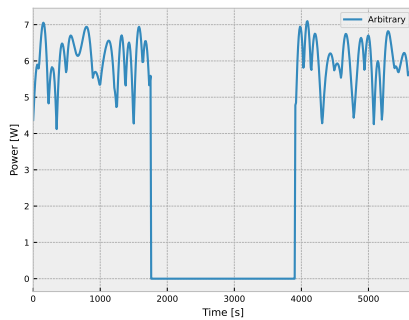
```

Algorithm 4: Function—solar_position

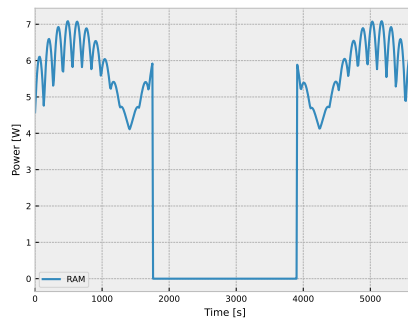
```

Input : day, month, year
Output:  $r_{\text{sun}}$ 
/* Compute Julian day */
1  $jd \leftarrow 367 \times \text{year} - \lfloor (7 \times (\text{year} + \lfloor (\text{month} + 9)/12 \rfloor) / 4 \rfloor + \lfloor 275 \times \text{month} / 9 \rfloor + \text{day} + 1721013.5;$  */
/* Astronomical unit (km) */
2  $AU \leftarrow 149597870.691$  */
/* Julian days since J2000 */
3  $n \leftarrow jd - 2451545$  */
/* Julian centuries since J2000 */
4  $cy \leftarrow n / 36525$  */
/* Mean anomaly (deg) */
5  $M \leftarrow 357.528 + 0.9856003 \times n$  */
6  $M \leftarrow \text{mod}(M, 360)$  */
/* Mean longitude (deg) */
7  $L \leftarrow 280.460 + 0.98564736 \times n$  */
8  $L \leftarrow \text{mod}(L, 360)$  */
/* Apparent ecliptic longitude (deg) */
9  $\lambda \leftarrow L + 1.915 \times \sin(M) + 0.020 \times \sin(2 \times M)$  */
10  $\lambda \leftarrow \text{mod}(\lambda, 360)$  */
/* Obliquity of the ecliptic (deg) */
11  $\epsilon \leftarrow 23.439 - 0.0000004 \times n$  */
/* Unit vector from earth to sun */
12  $u \leftarrow [\cos(\lambda); \sin(\lambda) \times \cos(\epsilon); \sin(\lambda) \times \sin(\epsilon)]$  */
/* Distance from earth to sun (km) */
13  $r_{\text{sun}} \leftarrow (1.00014 - 0.01671 \times \cos(M) - 0.000140 \times \cos(2 \times M)) \times AU$  */
/* Geocentric position vector (km) */
14  $r_{\text{sun}} \leftarrow r_{\text{sun}} \times u$ 

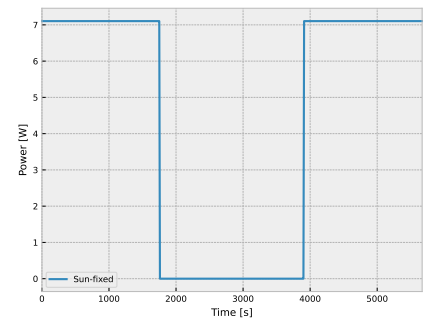
```



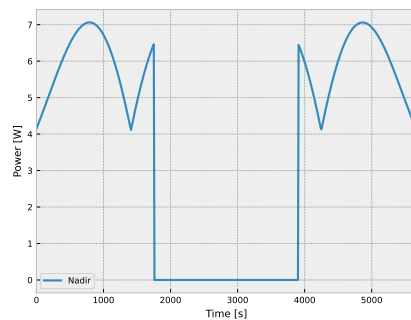
(a) Arbitrary



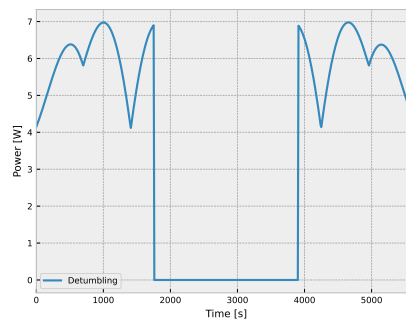
(b) RAM



(c) Sun-fixed



(d) Nadir



(e) Magnet

Figure 2. Examples of power generation for different attitudes.**3.2. General Procedure for Scheduling Data**

The main objective here is to generate an ONTS instance based on random but realistic data for any mission size and orbit duration. The instances can be made more difficult to schedule on demand by increasing the number of jobs or time units to consider. For that we created the instance generator procedure presented in Algorithm 5. It takes two inputs:

the number of jobs (J) and the number of time units to be considered (T). The output of the algorithm is the values for nine variables for each job: u_j , q_j , y_j^{\min} , y_j^{\max} , t_j^{\min} , t_j^{\max} , p_j^{\min} , p_j^{\max} , w_j^{\min} , and w_j^{\max} . This fully characterizes an instance for the ONTS regarding the QoS aspects.

Algorithm 5: Instance generator algorithm

Input : Number of jobs J , number of time periods T
Output: Initial values for $u_j, q_j, y_j^{\min}, y_j^{\max}, t_j^{\min}, t_j^{\max}, p_j^{\min}, p_j^{\max}, w_j^{\min}, w_j^{\max}$

```

1 for  $j \leftarrow 1$  to  $J$  do
2    $u_j \leftarrow U(1, J)$ ;
3    $q_j \leftarrow U(0.3, 2.5)$ ;
4    $y_j^{\min} \leftarrow U[1, \lceil T/45 \rceil]$ ;
5    $y_j^{\max} \leftarrow U[y_j^{\min}, \lceil T/15 \rceil]$ ;
6    $t_j^{\min} \leftarrow U[1, \lceil T/10 \rceil]$ ;
7    $t_j^{\max} \leftarrow U[t_j^{\min}, \lceil T/4 \rceil]$ ;
8    $p_j^{\min} \leftarrow U[t_j^{\min}, \lceil T/4 \rceil]$ ;
9    $p_j^{\max} \leftarrow U[p_j^{\min}, T]$ ;
10   $w_j^{\min} \leftarrow U[0, \lceil T/5 \rceil]$ ;
11   $w_j^{\max} \leftarrow U[\lceil T - \lceil T/5 \rceil, T]$ ;
12 end
  
```

4. Data Validation

A few studies have already been conducted and published using the instance generation methodology presented here—although never previously explained in detail—demonstrating they are realistic and representative of the problem [4–8]. In [4], for instance, the authors propose a set of instances for the ONTS problem [18]. These instances were designed to evaluate the performance of their branch-and-price (B&P) method for solving the ONTS problem. The cases are based on the FloripaSat-I mission, which is a nanosatellite with a 628 km altitude and a 97 minute orbit period. The amount of time units T varies, where for values greater than 97 the extra units indicate alternative orbits or a finer time resolution—for example, scheduling in 30 second time steps. Furthermore, because the number of tasks directly correlates with the activation and deactivation timings of payloads on a nanosatellite, they reflect varying spacecraft sizes given that a larger volume may accommodate more payloads.

We now present the results of applying [4] scheduling approach on data generated using the methodology presented here. First, the orbital characteristics of the FloripaSat-I mission, as presented in Table 3, were used as input data for the power harvesting computation, and the power available for tasks was obtained by multiplying the power vector by an Electrical Power System (EPS) efficiency of 0.85.

Table 3. TLE input for the case study.

CubeSat	ω	i	e	M_m	M	Ω
FloripaSat-I	111.38	97.95	0.0016	14.82	248.91	225.78

Then, a random instance was generated using Algorithm 5 with $J = 9$ and $T = 170$ as input. The output parameters are shown in Table 4.

Table 4. Nanosatellite jobs data.

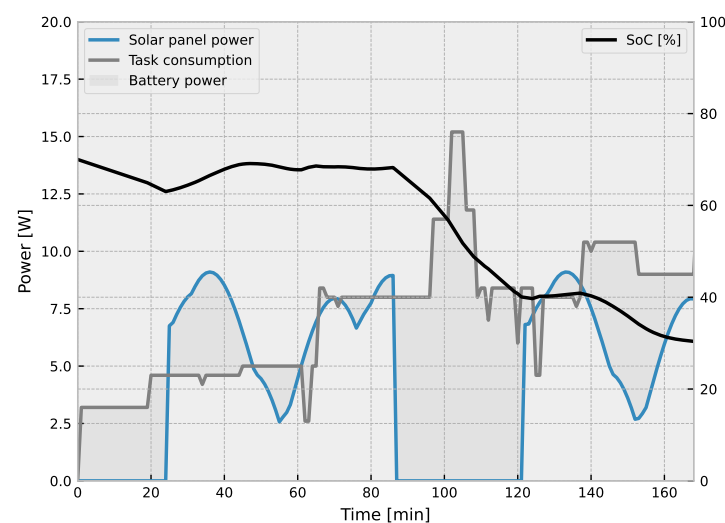
j	0	1	2	3	4	5	6	7	8
u_j	5	4	7	5	6	4	2	5	6
q_j	2.4	0.4	3.4	0.4	3.4	1.4	2.4	0.4	3.4
y_j^{\min}	1	2	1	1	1	2	1	1	2
y_j^{\max}	12	17	18	19	1	3	15	7	10
t_j^{\min}	6	14	27	47	23	30	41	4	9
t_j^{\max}	129	23	43	140	35	116	144	37	64
p_j^{\min}	18	57	51	58	34	41	57	4	53
p_j^{\max}	133	167	154	140	103	117	139	39	140
w_j^{\min}	1	1	1	1	1	20	1	1	1
w_j^{\max}	T	125	T	T	T	153	T	T	T

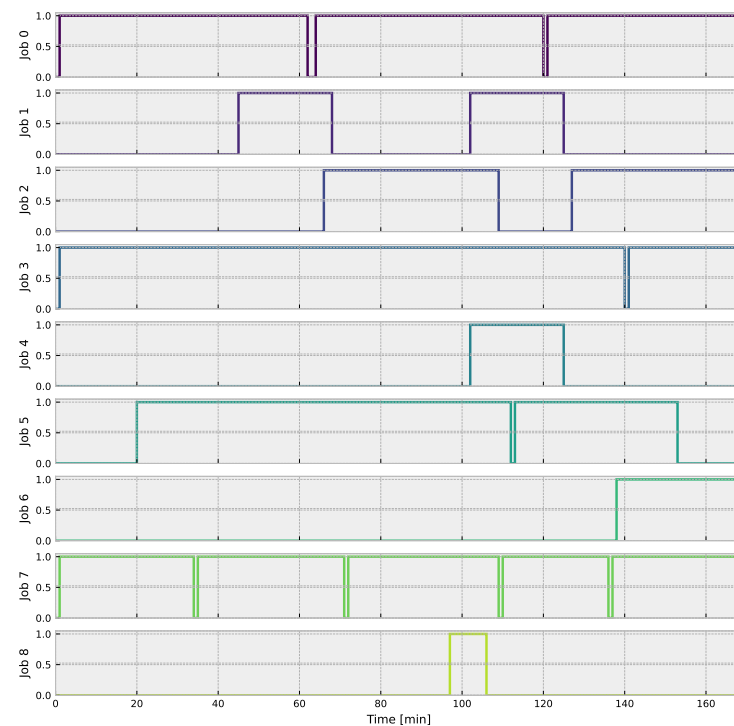
Finally, we attributed values to the battery and power management system parameters considering a 3U size nanosatellite, as presented in Table 5.

Table 5. Scheduling parameters.

Q	V_b	e_{cd}	SoC_0	ρ	γ
5 Ah	3.6 V	0.9	70%	0.3	5 A

Figure 3 shows the optimal schedule calculated for this example and the power analysis. The battery behavior along the orbit is shown in black, representing the State-of-Charge (SoC) at each moment. Notice that the battery power in gray can be power going to or coming from the battery, depending on the energy balance. The power input available to the tasks is shown in blue, including the eclipse time (0 W). One can notice how the battery SoC drops in eclipse time, where there is also an increase in tasks running, consuming more power. Figure 3b shows tasks from 0 to 8 where the x-axis goes from 0 to time T , and the y-axis shows whether the task is on or off for each unit of time. When comparing this scheduling with parameters of Table 4 it can be seen that they were followed accordingly. Job 5, for instance, cannot run before time 30 (w_j^{\min}) and after time 153 (w_j^{\max}), and so it was not.

**(a)****Figure 3.** Cont.



(b)

Figure 3. (a) Power (W) and SoC (%) usage along the orbit; (b) schedule for each job.

5. Conclusions

In conclusion, this work offers a collection of cases of the Offline Nanosatellite Task Scheduling (ONTS) optimization problem. These instances were built using genuine input data based on the orbital characteristics of the FloripaSat-I mission and power harvesting calculations. They were created primarily to benchmark the performance of different solution approaches for the ONTS issue. The cases are intended to be scalable in size and complexity. The authors emphasize the need to share examples to ensure scientific research repeatability and provide a baseline for benchmarks in the ONTS problem.

Furthermore, this research emphasizes the crucial role of power budget prediction for successful nanosatellite mission planning and satellite job scheduling. Power budget estimation helps engineers make educated decisions about the number and type of payloads that may be put on the satellite, as well as offline scheduling algorithms to obtain the best times to activate and deactivate payloads, optimizing the use of available energy and mission extraction of value.

In other words, this study advances the area of nanosatellite scheduling by offering a collection of publicly available examples for researchers to utilize and compare their results against, as well as stressing the relevance of power budget prediction for optimal mission planning. As CubeSat missions are planned based on the data from our methodology, future work should be directed at validating and refining both the data and methods presented here.

Author Contributions: Writing—original draft preparation, C.A.R., E.M.F. and L.O.S.; writing—review and editing, L.L. and V.R.Q.L.; software, C.A.R., E.M.F. and L.O.S.; methodology, C.A.R., E.M.F. and L.O.S.; conceptualization, L.O.S.; supervision, L.O.S.; funding acquisition, L.L. and V.R.Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by national funds through the Foundation for Science and Technology, I.P. (Portuguese Foundation for Science and Technology) by the project UIDB/05064/2020 (VALORIZA—Research Center for Endogenous Resource Valorization). The authors also acknowledge support from CNPq under grant numbers 404576/2021-4 and 150281/2022-6 and from FAPESC under grant number 2021TR001851.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data to support the results presented in this paper and the proposed model are available at: <https://github.com/c-a-rigo/ONTS-data> (accessed on 29 January 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ONTS	Offline Nanosatellite Task Scheduling
QoS	Quality of Service
EPS	Electrical Power System
B&P	Branch-and-Price
SoC	State-of-Charge

Symbols

The following symbols are used in this manuscript:

J	number of tasks.
T	number of time units.
u_j	priority of task j .
t_j^{\min} / t_j^{\max}	minimum and maximum running time required by task j .
y_j^{\min} / y_j^{\max}	minimum and maximum startups required by task j .
p_j^{\min} / p_j^{\max}	minimum and maximum period required by task j .
w_j^{\min} / w_j^{\max}	start and finish time of a task j execution window.
V_b	battery voltage, in Volts.
Q	battery capacity, in Ampere-hour.
ρ	battery level lower bound.
e_{cd}	battery charge/discharge efficiency.
γ	upper bound for battery discharge current, in Ampere.
r_t	energy harvested and available for tasks at time t .
q_j	power drained running task j .
i	orbit inclination in $^{\circ}$.
Ω	right ascension of the ascending node in $^{\circ}$.
e	eccentricity [-].
ω	argument of perigee in $^{\circ}$.
M	mean anomaly in $^{\circ}$.
M_m	mean motion in revolutions <i>per</i> day.
P	power generation.
η	power efficiency.
G	solar flux.
A	solar cell's area.
F	view factor.
ξ	eclipse function.
\mathbf{r}_X	position at the equatorial geocentric frame of reference.
$\mathbf{r}_{\bar{x}}$	position at the perifocal frame of reference.
\mathbf{Q}	Euler angle sequence.
h	specific relative angular momentum.
θ	true anomaly.

μ	gravitational parameter.
E	eccentric anomaly.
\mathbf{r}_{sun}	position of the sun.
\mathbf{n}	initial orientation.
$\mathbf{N}_{\bar{x}}$	orientation at the perifocal frame of reference.
\mathbf{N}_X	final orientation at the equatorial geocentric frame of reference.
\mathbf{B}	matrix of rotation.

References

- Villela, T.; Costa, C.A.; Brandão, A.M.; Bueno, F.T.; Leonardi, R. Towards the thousandth CubeSat: A statistical overview. *Int. J. Aerosp. Eng.* **2019**, *2019*, 5063145. [CrossRef]
- Shkolnik, E.L. On the verge of an astronomy CubeSat revolution. *Nat. Astron.* **2018**, *2*, 374–378. [CrossRef]
- Poghosyan, A.; Golkar, A. CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions. *Prog. Aerosp. Sci.* **2017**, *88*, 59–83. [CrossRef]
- Rigo, C.A.; Seman, L.O.; Camponogara, E.; Morsch Filho, E.; Bezerra, E.A.; Munari, P. A branch-and-price algorithm for nanosatellite task scheduling to improve mission quality of service. *Eur. J. Oper. Res.* **2022**, *303*, 168–183. [CrossRef]
- Rigo, C.A.; Seman, L.O.; Camponogara, E.; Filho, E.M.; Bezerra, E.A. A nanosatellite task scheduling framework to improve mission value using fuzzy constraints. *Expert Syst. Appl.* **2021**, *175*, 114784. [CrossRef]
- Rigo, C.A.; Seman, L.O.; Camponogara, E.; Morsch Filho, E.; Bezerra, E.A. Task scheduling for optimal power management and quality of service assurance in CubeSats. *Acta Astronaut.* **2021**, *179*, 550–560. [CrossRef]
- Seman, L.O.; Ribeiro, B.F.; Rigo, C.A.; Filho, E.M.; Camponogara, E.; Leonardi, R.; Bezerra, E.A. An Energy-Aware Task Scheduling for Quality-of-Service Assurance in Constellations of Nanosatellites. *Sensors* **2022**, *22*, 3715. [CrossRef] [PubMed]
- Camponogara, E.; Seman, L.O.; Rigo, C.A.; Morsch Filho, E.; Ribeiro, B.F.; Bezerra, E.A. A continuous-time formulation for optimal task scheduling and quality of service assurance in nanosatellites. *Comput. Oper. Res.* **2022**, *147*, 105945. [CrossRef]
- Marcelino, G.M.; Morsch Filho, E.; Martinez, S.V.; Seman, L.O.; Bezerra, E.A. In-orbit preliminary results from the open-source educational nanosatellite FloripaSat-I. *Acta Astronaut.* **2021**, *188*, 64–80. [CrossRef]
- Marcelino, G.M.; Vega-Martinez, S.; Seman, L.O.; Kessler Slongo, L.; Bezerra, E.A. A Critical Embedded System Challenge: The FloripaSat-1 Mission. *IEEE Lat. Am. Trans.* **2020**, *18*, 249–256. [CrossRef]
- Ning, G.; Haran, B.; Popov, B.N. Capacity fade study of lithium-ion batteries cycled at high discharge rates. *J. Power Sources* **2003**, *117*, 160–169. [CrossRef]
- Li, J.; Gee, A.M.; Zhang, M.; Yuan, W. Analysis of battery lifetime extension in a SMES-battery hybrid energy storage system using a novel battery lifetime model. *Energy* **2015**, *86*, 175–185. [CrossRef]
- Gilmore, D.; Donabedian, M. *Spacecraft Thermal Control Handbook: Fundamental Technologies*; Spacecraft Thermal Control Handbook; Aerospace Press: El Segundo, CA, USA, 2002.
- Filho, E.M.; Seman, L.O.; Rigo, C.A.; Nicolau, V.P.; Ovejero, R.G.; Leithardt, V.R.Q. Irradiation Flux Modelling for Thermal–Electrical Simulation of CubeSats: Orbit, Attitude and Radiation Integration. *Energies* **2020**, *13*, 6691. [CrossRef]
- Vega Martinez, S.; Filho, E.M.; Seman, L.O.; Bezerra, E.A.; Nicolau, V.d.P.; Ovejero, R.G.; Leithardt, V.R.Q. An Integrated Thermal–Electrical Model for Simulations of Battery Behavior in CubeSats. *Appl. Sci.* **2021**, *11*, 1554. [CrossRef]
- Curtis, H.D. *Orbital Mechanics for Engineering Students*, 3rd ed.; Butterworth-Heinemann: Boston, MA, USA, 2014; p. 751. [CrossRef]
- Walter, U. Spacecraft Attitude Dynamics. In *Astronautics: The Physics of Space Flight*; Springer International Publishing: Cham, Switzerland, 2018; pp. 699–734. [CrossRef]
- Rigo, C.A.; Seman, L.O.; Filho, E.M. Nanosatellite Scheduling Instances. 2023. Available online: <https://github.com/c-a-rigo/nanosatellite-scheduling-instances> (accessed on 27 February 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.