UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS DE FLORIANÓPOLIS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Cezar Antônio Rigo

Task scheduling for optimal power management and quality-of-service assurance in nanosatellites

Florianópolis
2022

Cezar Antônio Rigo

Task scheduling for optimal power management and quality-of-service assurance in nanosatellites

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina para a obtenção do título de doutor em Engenharia Elétrica.
Orientador: Prof. Eduardo Augusto Bezerra, Ph.D.
Coorientador: Prof. Laio Oriel Seman, Dr.

Florianópolis
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Rigo, Cezar Antônio
Task scheduling for optimal power management and quality-of-service assurance in nanosatellites / Cezar Antônio Rigo ; orientador, Eduardo Augusto Bezerra, coorientador, Laio Oriel Seman, 2022.
102 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Elétrica, Florianópolis, 2022.

Inclui referências.

1. Engenharia Elétrica. 2. Escalonamento de tarefas. 3. Otimização. 4. CubeSat. 5. Qualidade de serviço. I. Bezerra, Eduardo Augusto. II. Seman, Laio Oriel. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Elétrica. IV. Título.

Cezar Antônio Rigo

Task scheduling for optimal power management and quality-of-service assurance in nanosatellites

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Pedro Augusto Munari Junior, Dr.
Universidade Federal de São Carlos

Prof. Marcus Vinicius Soledade Poggi de Aragao, Ph.D.
Pontifícia Universidade Católica do Rio de Janeiro

Prof. Reinaldo Morabito Neto, Dr.
Universidade Federal de São Carlos

Prof. Rafael de Santiago, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de doutor em Engenharia Elétrica.

ICP-Edu
Documento assinado digitalmente
Telles Brunelli Lazzarin
Data: 13/09/2022 17:08:51-0300
CPF: 023.400.719-25
Verifique as assinaturas em https://v.ufsc.br

Prof. Telles Brunelli Lazzarin, Dr.
Coordenador do Programa

ICP-Edu
Documento assinado digitalmente
Eduardo Augusto Bezerra
Data: 12/09/2022 11:23:15-0300
CPF: 830.851.577-00
Verifique as assinaturas em https://v.ufsc.br

Prof. Eduardo Augusto Bezerra, Ph.D.
Orientador

Florianópolis, 12 de setembro de 2022.

Ao cosmos.

AGRADECIMENTOS

Antes de tudo, agradeço aos meus pais, Clezio e Ivone Rigo, a minha irmã, Angélica Rigo, e aos meus avós, pelo incondicional apoio durante estes três anos de dedicação integral aos estudos.

Agradeço ao professor Eduardo Augusto Bezerra pela oportunidade de trabalhar sob sua orientação. Sou muito grato ao meu coorientador Laio Oriel Seman por seu valioso direcionamento e feedback, sempre à disposição para contribuir e aperfeiçoar o trabalho. Igualmente, agradeço ao professor Eduardo Camponogara por suas inúmeras contribuições.

Agradeço aos meus queridos amigos Felipe Andreis, Pedro Rangel, Gonzalo Nequesaurt, Daniele Prozczinski e Fábio Marques pela amizade, apoio e incentivo.

Por fim, agradeço aos brasileiros que por meio de seu trabalho e honestidade, financiam ensino e pesquisa de qualidade nas universidades Federais.

“The true delight is in the finding out
rather than in the knowing.”
(Isaac Asimov)

RESUMO

O escalonamento de tarefas é uma abordagem eficaz para aumentar a extração de valor de uma missão espacial, podendo proporcionar uma melhor gestão de recursos e garantias de qualidade de serviço. Apesar de estar sujeito a muitas restrições, o escalonamento de tarefas em um satélite é, em última análise, restringido pela quantidade de energia disponível a qualquer momento. Nesta tese, é desenvolvido um *framework* para escalonamento de tarefas em nanossatélites focado no gerenciamento ótimo da energia disponível, enquanto garantindo qualidade de serviço. Uma formulação matemática de programação inteira (IP), projetada para maximizar o número de tarefas a serem executadas por um satélite, restrita à quantidade de potência disponível, momento a momento, ao longo de uma órbita é apresentada. O modelo de otimização é formulado para contemplar a prioridade da tarefa, número mínimo e máximo de ativação, tempo mínimo e máximo de execução, período mínimo e máximo e janela de execução. A decomposição de Dantzig-Wolfe é usada para explorar a estrutura especial da formulação (MILP), decompondo-a por tarefas, o que resulta em uma nova formulação baseada em colunas para o problema. Para resolver a formulação resultante, é proposto um algoritmo *branch-and-price* (B&P) adequado para o escalonamento de um grande número de tarefas em um horizonte de tempo expandido. O vetor variante de energia de entrada foi calculado com base na eficiência das células solares e em um modelo analítico usado para estimar o campo de irradiância segundo parâmetros de órbita e atitude. Para demonstrar a aplicabilidade da metodologia, vários experimentos foram conduzidos considerando quatro tamanhos de satélites com diferentes órbitas e parâmetros de tarefa. Os resultados mostram que a estratégia de escalonamento *offline* proposta gera um plano de escalonamento eficiente e ótimo, permitindo o melhor uso possível dos recursos energéticos disponíveis e garantindo a qualidade de serviço. Além disso, a metodologia de solução B&P se mostrou 88% mais eficiente computacionalmente para chegar a escalonamentos ótimos contraposto ao MILP.

Palavras-chave: Escalonamento de tarefas. Otimização. Modelo de bateria. CubeSat. Qualidade de serviço.

RESUMO EXPANDIDO

Introdução

Estabelecido pela Universidade Politécnica Estadual da Califórnia em 1999 e com adoção significativa desde então, o padrão CubeSat, ou nanossatélite, é caracterizado como um cubo de borda de 10 centímetros, ou 1U, com peso máximo de 1,33 kg e empilhamento máximo de dezesseis unidades, ou 16U. Com mais de 1.500 nanossatélites lançados atualmente, o sucesso do formato é atribuído à produção em massa de peças e ao uso de componentes comerciais de prateleira que permitem um rápido tempo de desenvolvimento – em última análise, às custas de uma vida útil mais curta. Adotado inicialmente para fins educacionais, recentemente, o formato vem sendo utilizado em um número crescente de aplicações comerciais e de alto valor agregado devido aos avanços da miniaturização de hardware e quase duas décadas de pesquisa e desenvolvimento. Apesar de anos de pesquisa e desenvolvimento, mais pesquisas são essenciais, especialmente sobre problemas decorrentes de uma limitação significativa deste padrão de espaçonave: a pequena área de superfície para painéis fotovoltaicos e os consequentes baixos níveis de captura de energia. Vários trabalhos abordaram essa questão com foco no hardware, resultando em melhorias de eficiência em painéis solares, armazenamento de energia e projeto de circuitos do Sistema Elétrico de Potência (EPS), e consequentemente gerando uma melhor extração do valor da missão. Uma abordagem complementar consiste na aplicação de técnicas de Pesquisa Operacional, que podem melhorar o gerenciamento de energia e, portanto, maximizar o valor da missão, por meio do planejamento inteligente da mesma e do escalonamento de tarefas. A maioria dos trabalhos da área considera missões de Observação da Terra (EO) com múltiplos satélites, com formulações resolvidas *online*, a bordo da espaçonave, resultando em tomadas de decisão não ótimas. Poucos trabalhos de pesquisa focam nas características específicas do escalonamento de tarefas dos nanossatélites, suas limitações e aplicações. O agendamento *online*, por exemplo, não é adequado para nanossatélites devido ao seu baixo poder de computação integrado e porque a otimização é desejada, pois os recursos são mais escassos. Os protocolos de agendamento de satélites são diretamente responsáveis pela quantidade de trabalho realizado em órbita; além disso, o algoritmo de escalonamento pode verificar se as tarefas podem ser executadas com a frequência desejada, ou se as cargas úteis devem ser reduzidas. Nesta tese, é desenvolvido um *framework* para escalonamento de tarefas em nanossatélites focado no gerenciamento ótimo da energia disponível, enquanto garantindo qualidade de serviço.

Objetivos

O objetivo da presente tese é abordar as duas principais lacunas de pesquisa identificadas e apresentadas no capítulo 2, que são: *(i)* analisar e modelar o problema de escalonamento de nanossatélites visando *(ii)* maximizar a utilização dos recursos energéticos e otimizar a extração de valor da missão. Especificamente: (1) compreender as aplicações, necessidades e requisitos específicos das missões de nanossatélites e como estão sendo abordados na literatura científica; (2) criar uma nova formulação matemática de programação linear inteira mista (MILP) genérica o suficiente para agendar prontamente qualquer missão ou tarefa em qualquer ou vários subsistemas de um satélite abordando totalmente o problema de agendamento de nanossatélites e suas especificidades; (3) elaborar uma metodologia de agendamento multi-órbita junto com um modelo de bateria, garantindo que a vida útil da bateria seja preservada e estendida; (4) desenvolver um algoritmo *branch-and-price* por meio da aplicação da decomposição de

Dantzig-Wolf e procedimento de geração de colunas (CG), explorando a estrutura específica da formulação MILP; (5) comparar e avaliar os resultados computacionais do *branch-and-price* com a metodologia base MILP.

Metodologia

Uma formulação matemática de programação inteira (MILP), projetada para maximizar o número de tarefas a serem executadas por um satélite, restrita à quantidade de potência disponível, momento a momento, ao longo de uma órbita é a metodologia base utilizada. O modelo de otimização é formulado para contemplar a prioridade da tarefa, número mínimo e máximo de ativação, tempo mínimo e máximo de execução, período mínimo e máximo e janela de execução. Um modelo de bateria é apresentado, com estratégias para extensão de sua vida útil: um programa disjuntivo é projetado para considerar as eficiências de carga e descarga da bateria, juntamente com restrições *fuzzy* para limitar as taxas de corrente de carga e descarga (taxas C e E) e profundidade de descarga (DOD). Cada acesso à bateria é penalizado na função objetivo, estimulando assim que o consumo de energia corresponda à energia de entrada. O vetor variante de energia de entrada foi calculado com base na eficiência das células solares e em um modelo analítico usado para estimar o campo de irradiância segundo parâmetros de órbita e atitude. Como a formulação MILP criada pode se tornar muito difícil de resolver por solucionadores de otimização de propósito geral quando instâncias de tamanho real são consideradas, uma abordagem mais eficaz é projetada explorando a estrutura especial da matriz de coeficientes da formulação. Para tanto, implementa-se aqui a decomposição de Dantzig-Wolfe, que possibilita o uso da técnica de geração de colunas para melhorar o desempenho. Essa decomposição divide a formulação original em um problema mestre e um ou mais subproblemas. O problema mestre considera apenas um subconjunto das restrições na formulação original, e é definido por variáveis relacionadas a pontos inteiros e raios do poliedro relacionados às restrições que não foram incluídas. A vantagem é que esses pontos e raios podem ser gerados iterativamente, recorrendo ao(s) subproblema(s), em um *framework* conhecido como técnica de geração de colunas (CG). Se a solução obtida no final do algoritmo CG for inteira, então ela também é ótima, caso contrário, o algoritmo é incorporado em uma estrutura *branch-and-bound* para encontrar a solução ótima inteira, um procedimento conhecido como *branch-and-price* (B&P).

Resultados

Os resultados do escalonamento da formulação de programação inteira (IP) inicial demonstram que esta abordagem gera um plano de missão com eficiência energética ideal, ao mesmo tempo em que satisfaz todos os requisitos das tarefas, permitindo a melhor captura e utilização de recursos energéticos e, além disso, garantindo QoS. Quando incluindo um modelo de bateria, criando um modelo MILP, pôde-se inferir que algumas vezes a formulação com a bateria é a única capaz de viabilizar o modelo, principalmente por que permite executar tarefas em momentos de eclipse solar na órbita. Além disso, a formulação gerencia melhor a energia disponível em órbita, maximizando a utilidade do satélite, em relação à metodologia inicial. Os resultados de experimentos computacionais envolvendo a geração de colunas com instâncias geradas aleatoriamente, porém realistas, mostraram uma melhoria de 70% no tempo total de solução em comparação com a abordagem padrão, que consiste em resolver um problema MILP com um *solver* comercial. Foram programados diversos exemplos ilustrativos baseados em missões realistas, cujos resultados corroboram a consistência da abordagem *branch-and-price* proposta. Além disso, uma redução significativa nas lacunas de otimalidade foi observada em todas as

instâncias ao usar a abordagem proposta, permitindo ao tomador de decisão obter rapidamente soluções quase ótimas para a melhor extração de valor da missão.

Discussão

A formulação de programação inteira (IP) inicial abordou o objetivo específico (2), estabelecendo uma metodologia de escalonamento genérica o suficiente para operar com diferentes resoluções de tarefas. A tarefa considerada pode ser uma execução feita por um dos sistemas embarcados (ou seja, uma tarefa do sistema operacional) ou mesmo a operação de uma carga útil. Portanto, a função desempenhada pela tarefa (ou seja, download de dados, *beacon*) ou pelo próprio satélite (ou seja, observação da Terra, rede de comunicações) é indiferente ao modelo de otimização criado, permitindo agendar prontamente qualquer tarefa. As posteriores modificações no modelo, resultando em um MILP, atenderam o objetivo (3), modelando uma bateria e formulando o acesso à energia como uma restrição *fuzzy*, permitindo que o escalonador exceda de maneira ótima (decidida pelo *solver* de otimização) o limite de energia previamente imposto pela entrada de energia dos painéis solares momento a momento. O benefício de considerar a bateria foi demonstrado no aumento do valor da soma de prioridade das tarefas quando comparado com o modelo sem bateria. Além disso, a bateria permite agendar tarefas críticas em períodos de tempo de eclipse (sem entrada de energia dos painéis solares). O escalonamento multi-órbita também foi uma adição importante, pois embora o regime cíclico apareça em alguns casos, a formulação multi-órbita serve para trabalhar com mudanças no nível de energia da bateria, que podem fazer com que os regimes periódicos de carga útil mudem ao longo do tempo para atingir o viabilidade do problema de otimização. Além de estar intimamente relacionado com a execução das tarefas, o nível da bateria é dependente da energia de entrada disponível no satélite. Em órbitas fechadas, a variação na entrada não é perceptível. No entanto, ao analisar a vida útil do satélite, pode haver uma variação significativa nos níveis de entrada de energia devido a distúrbios em órbita. A estratégia de decomposição e ramificação de Dantzig-Wolfe, por sua vez, permitiu um escalonamento para granularidade de tempo mais fina e maior quantidade de tarefas, completando o objetivo (4). Finalmente, as comparações visando atender ao objetivo (5) demonstraram que a metodologia *branch-and-price* é, em média, 88% mais rápida em obter resultados ótimos contraposto aos tempos computacionais do MILP resolvido com *solver* comercial.

Considerações Finais

A ideia de estimar um cronograma de tarefas pode servir de base para um projeto de nanossatélites. Com base em seus resultados, o engenheiro, por exemplo, pode optar por incluir ou remover uma carga útil, proibir a execução de tarefas simultaneamente, estimar a ocorrência de modos seguros de operação, considerar o intervalo de tempo entre o lançamento e a operação nominal plena, cenários de uso, e assim por diante. O trabalho apresentado nesta tese abordou este problema de escalonamento, atendendo sete necessidades específicas, conforme indicado nos objetivos. Em conclusão, acredita-se que uma estrutura robusta e realista para o agendamento *offline* ótimo de missões de nanossatélites tenha sido alcançada, o que capacita o tomador de decisão a agendar prontamente qualquer missão em um ou vários subsistemas de um nanossatélite.

Palavras-chave: Escalonamento de tarefas. Otimização. Modelo de bateria. CubeSat. Qualidade de serviço.

ABSTRACT

Task scheduling is an effective approach to increase the value of a satellite mission, which leads to improved resource management and quality of service. Despite subject to many constraints, satellite task scheduling is ultimately restricted by the amount of power available at any given moment. In this thesis, a nanosatellite task scheduling framework for optimal power management and quality-of-service assurance is developed. A mixed-integer linear programming (MILP) formulation is proposed, designed to maximize the number of tasks to be executed by a satellite, constrained to the amount of power available at any moment along the course of an orbit is presented. The optimization model is formulated to contemplate task priority, minimum and maximum number of task activation, minimum and maximum execution time, minimum and maximum period of a given task and execution window. The Dantzig-Wolfe decomposition is used to explore the special structure of the (MILP) formulation, decomposing it by tasks, which results in a novel profile-based formulation for the problem. To solve the resulting formulation, a branch-and-price (B&P) algorithm is developed, that is suitable for the scheduling of a large number of tasks over an expanded time horizon. The variant power input vector was calculated based on the solar cells efficiency and on an analytical model used to estimate the irradiance field according to parameters of orbit and attitude. To demonstrate the applicability of the methodology several experiments were conducted considering four satellite sizes with different orbits and task parameters. The results show that the proposed offline scheduling algorithm generates an optimal energy effective scheduling plan, allowing the best possible use of available energy resources while ensuring the quality of service (QoS). Furthermore, the B&P methodology proved to be 88% more computationally efficient to reach optimal schedules as opposed to the MILP.

Keywords: Task scheduling. Optimization. Battery model. CubeSat. Quality of service.

FIGURE LIST

Figure 1 – A 2U nanosatellite and its modules. 21
Figure 2 – Examples of a nanosatellite's attitude. 23
Figure 3 – Filtering process. 28
Figure 4 – 1U simulation results 45
Figure 5 – 2U simulation results 47
Figure 6 – 3U simulation results 48
Figure 7 – Impact of $\mathcal{T}$ size on problem-solving time. 49
Figure 8 – Orbit diagram. 58
Figure 9 – Fraction of eclipse for FloripaSat-I orbit along a year. 60
Figure 12 – β variation impact on task priority sum – cost function (2). 64
Figure 14 – Power balance in scenarios A and B. 66
Figure 15 – SoC histogram. 67
Figure 16 – SoC analysis for 20 consecutive orbits and different β values. $\overline{OBJ}$ is the average objective value per orbit and $\overline{ST}$ is the average solving time per orbit. 67
Figure 18 – Illustration of branching tree. 76
Figure 21 – Instance #14 gap evolution for Gurobi and B&P. 85
Figure 24 – SoC histogram for 3U scenario 89
Figure 25 – γ variation impact on the objective value – see expression (35a). 89

TABLE LIST

Table 1 – Research questions. 27
Table 2 – Search terms. 27
Table 3 – Filtering criteria. 28
Table 4 – Strategies being used to formulate the satellite scheduling problem. 30
Table 5 – Type of satellite missions formulated. 31
Table 6 – Type of satellite modeled. 31
Table 7 – Algorithms being used to solve the scheduling formulations. 32
Table 8 – Scheduling modes. 33
Table 9 – Energy in the satellite scheduling formulations. 34
Table 10 – Sets, indices and decision variables. 40
Table 11 – Parameters. 41
Table 12 – 1U payloads scheduling data. 45
Table 13 – 2U nanosatellite payloads scheduling data. 46
Table 14 – 3U nanosatellite payloads scheduling data. 47
Table 15 – Added sets, constants and variables. 51
Table 16 – TLE input for the case studies. 59
Table 17 – Nanosatellite jobs data. 61
Table 18 – Scheduling parameters. 61
Table 19 – Nanosatellite task data. 68
Table 20 – Scheduling parameters. 68
Table 21 – Added sets, indexes, variables and constants. 73
Table 22 – Orbital parameters. 80
Table 23 – Standard deviation between columns for each job. 83
Table 24 – Performance of the multiple-column B&P algorithm *vs.* Mixed Integer Linear Programming (MILP) formulation using the general-purpose solver Gurobi for 21 instances (time limit = 7200 seconds). 84
Table 25 – Nanosatellite tasks data (3U mission). 86
Table 26 – Orbit and battery parameters (3U mission). 86
Table 27 – Nanosatellite task data (6U mission). 89
Table 28 – Scheduling parameters (6U mission). 90

LIST OF ACRONYMS

CG	Column Generation
COTS	Commercial Off The Shelf
CPU	Central Processing Unit
CSPM	Constraint Satisfaction Problem Model
EDF	Early Deadline First
EO	Earth Observation
EPS	Electrical Power System
FLP	Fuzzy Linear Programming
GDP	Generalized Disjunctive Program
GTM	Graph Theory Model
IP	Integer Programming
LB	Lower Bound
LEO	Low Earth Orbit
LP	Linear Programming
MILP	Mixed Integer Linear Programming
ML	Machine Learning
MP	Master Problem
MPM	Mathematical Programming Model
MPPT	Maximum Power Point Tracking
OBDH	On-Board Data Handler
PC	Personal Computer
PCB	Printed Circuit Board
QoS	Quality of Service
RADAR	Radio Detection And Ranging
RAM	Random Access Memory
RMP	Restricted Master Problem
SoC	State of Charge
TLE	Two Line Element
UFSC	Federal University of Santa Catarina

GLOSSARY

q_{sj}	Power consumed by executing a task j on subsystem s.
r_t	Amount of power input at time t (from solar panels).
S	Number of subsystems.
J	Number of tasks.
T	Number of time periods.
u_{sj}	Priority of task j on subsystem s.
$t_{sj}^{\min}$	Minimum CPU time of task j on subsystem s.
$t_{sj}^{\max}$	Maximum CPU time of task j on subsystem s.
$y_{sj}^{\min}$	Minimum number of startups of task j on subsystem s.
$y_{sj}^{\max}$	Maximum number of startups of task j on subsystem s.
$p_{sj}^{\min}$	Minimum period of task j on subsystem s.
$p_{sj}^{\max}$	Maximum period of task j on subsystem s.
$w_{sj}^{\min}$	Start moment for the execution window of task j on subsystem s.
$w_{sj}^{\max}$	Finish moment for the execution window of task j on subsystem s.
V_b	Battery nominal voltage.
SoC_t	Battery state of charge at time t.
e_c	Charge efficiency.
e_d	Discharge efficiency.
Q	Battery nominal capacity (Ah).
ρ	Minimum acceptable battery level.
b_t	Decision variable that takes the value of 1 each time the charging rate should be used and 0 each time the discharging rate should be used.
α_t	Decision variable that takes values between 0 and 1 representing the amount of energy drawn from the battery.
k_t	Difference between power input and power consumption at time t (W).
i_t	Battery current (A).
a_t	Assumes the values of $\lvert i_t \rvert$.
$c_j^k > 0$	Total priority associated to the k^{th} profile of task j.
$q_j^k(t) > 0$	Power usage associated to the k^{th} profile of task j at time t.

CONTENTS

1	**INTRODUCTION**	**19**
1.1	THESIS OBJECTIVES	20
1.2	THE NANOSATELLITE SCHEDULING PROBLEM	20
1.3	DOCUMENT STRUCTURE	24
1.4	PUBLICATIONS	24
2	**STATE OF THE ART**	**26**
2.1	METHODOLOGY	26
2.1.1	**Research questions**	**26**
2.1.2	**Research process**	**26**
2.1.3	**Publications filtering**	**27**
2.2	RESULTS	29
2.2.1	**GQ1 – What strategies are being used to formulate the task scheduling problem in satellites?**	**29**
2.2.2	**GQ2 – What types of missions are being modeled?**	**30**
2.2.3	**GQ3 – What types of satellites are being modeled?**	**30**
2.2.4	**GQ4 – What algorithms are being used to solve the formulations?**	**32**
2.2.5	**SQ1 – How are the strategies being applied in the task scheduling problem in satellites?**	**33**
2.2.6	**SQ2 – How is optimization being used to address the scheduling problem?**	**33**
2.2.7	**SQ3 – How is energy being modeled in these formulations?**	**34**
2.2.8	**PQ1 – Where have the works been published?**	**34**
2.2.9	**PQ2 – What is the average number of citations?**	**34**
2.3	DISCUSSION	35
2.4	CONCLUSIONS	37
3	**BASE MODEL FORMULATION**	**39**
3.1	POWER INPUT MODEL	39
3.2	OPTIMAL SCHEDULING ALGORITHM	40
3.2.1	**Solving the optimization problem**	**43**
3.3	RESULTS AND ANALYSIS	43
3.3.1	**Power input estimation**	**44**
3.3.2	**Optimization results**	**44**
3.3.2.1	1U Nanosatellite	44
3.3.2.2	2U Nanosatellite	46
3.3.2.3	3U Nanosatellite	46
3.3.3	**Computational impact**	**47**
3.4	DISCUSSION	49
4	**EXTENDED MULTI-ORBIT AND BATTERY FORMULATION**	**50**

4.1 FUZZY CONSTRAINTS . . . 50
4.1.1 Zimmermann's method . . . 51
4.1.2 Werners's method . . . 52
4.1.3 Tanaka's method . . . 52
4.1.4 Verdegay's method . . . 52
4.2 PROPOSED BATTERY MODEL . . . 53
4.3 ORBIT COUPLING . . . 55
4.3.1 Solving the optimization problem . . . 57
4.4 POWER INPUT . . . 57
4.4.1 Orbit model . . . 58
4.4.1.1 Attitude Model . . . 58
4.4.2 Studied scenarios . . . 59
4.5 RESULTS AND ANALYSIS . . . 60
4.5.1 Requisites: 3U mission . . . 60
4.5.1.1 Scenario A – 1/2 Max. Orbit . . . 61
4.5.1.2 Scenario B – Max. Orbit . . . 64
4.5.1.3 Results analysis . . . 66
4.5.2 Requisites: 6U mission . . . 68
4.5.2.1 Scenario C – Max. Orbit . . . 68
4.6 DISCUSSION . . . 69
5 BRANCH-AND-PRICE FORMULATION . . . 71
5.1 DANTZIG-WOLFE DECOMPOSITION AND COLUMN GENERATION ALGORITHM . . . 72
5.1.1 Master problem . . . 73
5.1.2 Column generation . . . 74
5.2 BRANCH-AND-PRICE . . . 75
5.2.1 Branching scheme . . . 75
5.2.2 Changes to Restricted Master Problem (RMP) and subproblems . . . 76
5.2.3 MIP heuristic . . . 77
5.2.4 Branch-and-price procedure . . . 77
5.3 COMPUTATIONAL EXPERIMENTS . . . 79
5.3.1 Power input calculation . . . 79
5.3.2 Initial evaluation . . . 80
5.3.2.1 Column generation . . . 81
5.3.3 Comparison with the traditional approach . . . 83
5.3.4 Illustrative examples . . . 85
5.3.4.1 3U Mission . . . 85
5.3.4.2 6U Mission . . . 88
5.4 DISCUSSION . . . 91

6 CONCLUSION 92
6.1 FURTHER RESEARCH DIRECTIONS 93
REFERENCES 94

1 INTRODUCTION

Nanosatellites, or CubeSats, are defined as cubic satellites of 10 cm x 10 cm x 10 cm, or "1U", with a maximum weight of 1.33 kg per U, and a maximum size of 16 Us (CHIN et al., 2008). This type of spacecraft is evolving from a purely low-cost educational and technology demonstration tool to a platform for a variety of high-quality scientific experiments and commercial applications (SHKOLNIK, 2018). This trend can be attributed to improvements in hardware miniaturization, along with more than 15 years of research and innovation. This prospect offers challenges and opportunities for the research community, companies, academics and national space agencies, particularly on making the technology more reliable and efficient.

Given such compact standard, capacity for energy harvesting is small compared to more traditional satellite formats, and consequently, energy management is even more crucial for mission success, and Quality of Service (QoS) guarantees. This management is usually achieved through scheduling, which consists of deciding the tasks that can be executed at any given moment considering service requirements and existing resources. Mission planning can also improve the extraction of value through better resources management, the completion of more tasks, assurances of adequate timing response, and fulfillment of overall task requirements.

Satellite scheduling protocols are directly responsible for the amount of work performed in orbit; nevertheless, despite being subject to many constraints and requirements, the ultimate limiting factor is the amount of energy available for consumption. In order to extract the most value of a mission, the scheduling algorithm goal should be to match energy consumption and energy input. That, however, could conflict with the task priorities and quality of service execution. Nevertheless, it is expected to exist a task plan for each satellite's attitude and orbit that ensures this energy-neutral condition, while maintaining the tasks execution requirements and QoS. The energy-neutral operation happens when energy consumption is less than or equal to energy input at all times.

The scheduling algorithm can run either *offline*, whereby the task execution plan determined a priori on the ground, or *online*, in real-time on-board the spacecraft. In this sense, an offline scheduling approach can help the engineer to estimate whether the satellite will be able to handle, for example, the desired amount of payloads. Thus, even subject to input power estimation errors, it can be verified whether tasks would be performed with the desired frequency or whether it would be necessary, for example, to reduce the number of payloads. When run offline, task scheduling can assist engineers estimate whether the satellite will handle, for example, the desired payloads. Thus, even subject to input power estimation errors, the scheduling algorithm can verify whether the tasks can be performed with the desired frequency, or whether the payloads have to be reduced.

Despite intense research in the field of satellite scheduling and increasing employment of nanosatellites in high-value missions, such as scientific experiments and commercial applications (POGHOSYAN; GOLKAR, 2017), almost no research has been directed to power management

and mission planning of CubeSats. This thesis presents an initial integer programming Integer Programming (IP) formulation aiming to generate a mission plan with optimal energy efficiency, while satisfying task requirements, allowing better capture and utilization of energy resources and, in addition, guaranteeing QoS. Subsequently, the base model is expanded to a Mixed Integer Linear Programming (MILP), including a battery model, to allow performing tasks at times of solar eclipse in orbit, and a method for scheduling multiple orbits. Then a decomposition and solution strategy based on column generation is formalized for the MILP, in an attempt to solve the problem for a larger time horizon or greater jobs number in a computational reasonable time.

Let us present this thesis objectives an then briefly discuss the complexities associated with planning a nanosatellite mission, which mostly arises from the shortcomings inherent to its format.

1.1 THESIS OBJECTIVES

The objective of the present thesis is to model the nanosatellite scheduling problem aiming at maximizing energy resources utilization and optimizing mission extraction of value. Specifically:

1. Understand the applications, necessities and requirements of nanosatellite missions and how they are being addressed in the scientific literature;
2. Formulate a novel MILP formulation generic enough for readily scheduling any mission or tasks in any or multiple subsystems of a satellite fully addressing the nanosatellite scheduling problem and its specificities;
3. Develop a multi-orbit scheduling methodology along with a battery model, ensuring that battery lifetime is preserved and extended;
4. Design a branch-and-price algorithm by means of applying the Dantzig-Wolf decomposition and column generation Column Generation (CG) procedure, exploring the specific structure of the MILP formulation;
5. Compare and evaluate the branch-and-price computational results with the base MILP methodology.

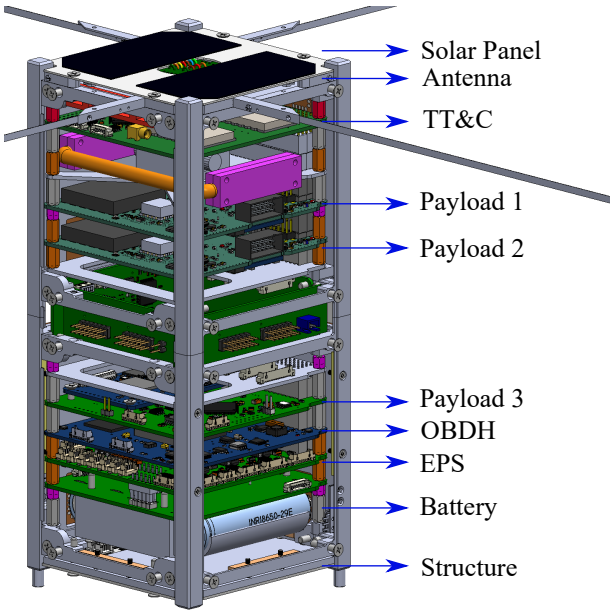
Therefore, this thesis aims to contribute to the state-of-the-art by presenting a novel, realistic and comprehensive nanosatellite scheduling framework, enabling any mission's best possible extraction of value.

1.2 THE NANOSATELLITE SCHEDULING PROBLEM

The service platform of a nanosatellite – illustrated in Figure 1 – usually has three main modules: (*i*) a Telemetry, Tracking and Command (TT&C) unit for communications; (*ii*) an

On-Board Data Handler (OBDH) for data processing; and (*iii*) the Electrical Power System (EPS) for energy conversion. The payloads, which effectively provide functionality and value to the spacecraft, are interfaced with the service platform and can be viewed as tasks of high importance and power impact to be scheduled. Therefore, energy management and internal space are critical constraints for how many payloads can be included and the overall mission extraction of value. For instance, a 2U size can typically accommodate six printed circuit boards, which could translate into three service modules and three payload modules, despite the energy harvesting capabilities being small, given the low surface area for PVs.

Figure 1 – A 2U nanosatellite and its modules.



Source: **The author**.

The tasks considered in this work refer to the activation and deactivation time of payloads, which are high resource impact components of nanosatellites and require careful management, given their critical importance for mission success. Therefore, in this problem, tasks are hardware modules integrated into the spacecraft, and their quantity depends on the nanosatellite size (*i.e.*, 1U, 2U, *etc.*), which does not change during the mission. These high energy impact tasks have to be pre-scheduled on Earth (*i.e.*, the ground station), offline, and later sent to the nanosatellite, which will leverage this information to define the order of tasks to be performed. It is considered that certain tasks can be part of subsystems, which could be aggregated to their specific needs. Therefore, the subsystems can refer to parts of the satellite or, in case of a single subsystem, the satellite itself.

During a mission conception, tasks typically have some specific behavior according to the payloads that they represent. First, it is considered that the tasks are non-preemptive, so, once started, they must execute for at least a minimum execution time, limited by a maximum

execution time. This gives the first requisite of the scheduling problem:

(*i*) Once a task has started, it must be completed successfully, contemplating its minimum and maximum execution time.

Consider, for example, a situation where a task is relative to data transmission. This task must be periodically transmitted every time the satellite passes over the ground station. This demand raises the second requisite:

(*ii*) Tasks are periodic within a given threshold, so a minimum and maximum time between startups of a given task are defined.

Also, it is necessary to ensure that tasks have a minimum number of activations to fulfill mission requirements. This number can also be limited to an upper bound if a task should execute no more than a given number of times:

(*iii*) The minimum and maximum number of startups within an orbit are defined for each task.

Through such requisites/properties, one can model, for instance, an environmental data acquisition routine, the capture of one or more images at a specific point in the orbit, or any other high-level activity performed by dedicated hardware such as payloads, sensors, cameras, and other modules with relevant impact on the energy consumption of the satellite. In addition, the tasks can have their ideal start and finish times optimally scheduled to guarantee the best possible mission value.

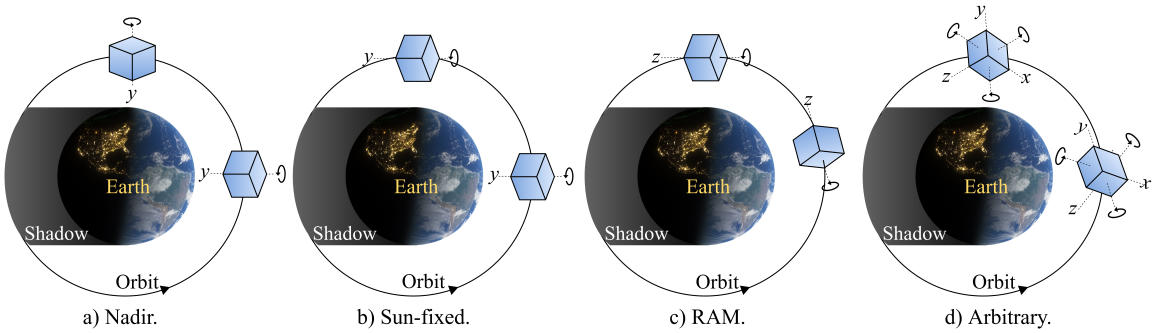
Other determinant aspects for energy capture are the orbital parameters, more significantly the time spent in the Earth's shadow, or eclipse time. A typical nanosatellite has no thrusters to adjust its trajectory and compensate for atmospheric drag and gravitational perturbations, resulting in an orbit deviation that leads to variations of the total eclipse time. For the FloripaSat-I mission developed by students at the Federal University of Santa Catarina (UFSC) and launched in 2019 (MARCELINO et al., 2020), the eclipse's fraction of the orbit (f_e) varies along the years from 0 to more than 35% of the orbit because the angle between the orbit plane and the Sun is not constant.

Although the solar radiation field at such a vast distance from the Sun can be considered parallel and approximated as a constant source of 1367 W/m^2 (GILMORE; DONABEDIAN, 2002), the spacecraft's kinematics can be quite convoluted. The faces of the nanosatellite towards the Sun at any given time, as well the angles of incoming radiation, have to be calculated precisely for an accurate energy harvesting prediction. In Figure 2, some typical nanosatellite attitudes are illustrated. For instance, the spacecraft can have the following rotations and axis pointing:

- Nadir: the satellite revolution is coupled to the orbit's period; therefore, one face (or axis) is permanently facing the Earth's surface. Attitude desirable for Earth Observation missions.

- Sun-fixed: the satellite keeps one of its axes permanently centered at the Sun, and therefore, one face (or axis) is permanently facing the Sun. Attitude useful for maximizing energy capture or to protect a critical component from radiation.
- Ram[1]: The axis of rotation is tangent to the trajectory, aligned with the satellite's velocity vector.
- Arbitrary spin: rotation is arbitrary in all axes, typical of a satellite without attitude control or damping forces.
- Magnet (not shown in Figure 2): the rotation alignment coincides with the Earth's magnetic field. A common attitude obtained by inserting a few magnets in the satellite's structure to promote this alignment.

Figure 2 – Examples of a nanosatellite's attitude.



Source: **The author**.

Clearly, each of these attitudes uniquely impacts the energy harvested and power management, hence the difficulty in planning a mission in an optimal way.

A realistic scheduling formulation needs not only to consider the energy being harvested at any instant but also the battery power and its dynamics, which raises the fourth requisite of the scheduling problem:

(*iv*) The battery state-of-charge must be considered, taking into consideration the energy harvesting capability of the nanosatellite.

For offline energy-constrained formulations, the mission planning can provide the developers valuable information on whether the mission's objectives align with the satellite's capabilities by optimally orchestrating the execution of the most energy-intensive tasks considering mission objectives, QoS requirements, battery charge, and energy harvesting.

[1] Name given due to the considered direction in which the satellite is ramming into a fluid, which in this case is the ionosphere.

1.3 DOCUMENT STRUCTURE

This thesis presents a compendium of papers or its sections published by the author during the research. To better present the author's contributions, this thesis is organized as follows:

- Chapter 2 presents a comprehensive review of related works and the state of the art. The text of this chapter was partially published as an article in Rigo, Seman, Camponogara, Morsch Filho, and Bezerra (2021);
- Chapter 3, published in Rigo, Seman, Camponogara, Morsch Filho, and Bezerra (2021), presents the methodology for the optimal scheduling formulation along with its results;
- Chapter 4 improves the formulation to allow multi-orbit scheduling and models the satellite's battery. This chapter was published in Rigo, Seman, Camponogara, Filho, et al. (2021b).
- Chapter 5 presents a Dantzig-Wolfe decomposition and branch-and-price strategy to solve bigger instances of the problem, specifically, to consider more tasks and finer time step. This chapter was published in Rigo, Seman, Camponogara, Morsch Filho, Bezerra, and Munari (2022);
- Lastly, Chapter 6 brings the final remarks and future work.

1.4 PUBLICATIONS

As mentioned in the previous Section, the work presented in this thesis has resulted in several publications, as listed bellow.

1. RIGO, Cezar Antônio; SEMAN, Laio Oriel; CAMPONOGARA, Eduardo; MORSCH FILHO, Edemar; BEZERRA, Eduardo Augusto. Task scheduling for optimal power management and quality-of-service assurance in CubeSats. **Acta Astronautica**, v. 179, p. 550–560, 2021. ISSN 0094-5765. DOI: 10.1016/j.actaastro.2020.11.016.
2. RIGO, Cezar A.; SEMAN, L. O.; CAMPONOGARA, E.; MORSCH FILHO, E., et al. Mission plan optimization strategy to improve nanosatellite energy utilization and tasks QoS capabilities. In: IV IAA Latin American Cubesat Workshop (IAA-LACW'2020). Paris: International Academy of Astronautics. [S.l.: s.n.], 2020.
3. RIGO, Cezar Antônio; SEMAN, Laio Oriel; CAMPONOGARA, Eduardo; FILHO, Edemar Morsch, et al. A nanosatellite task scheduling framework to improve mission value using fuzzy constraints. **Expert Systems with Applications**, v. 175, p. 114784, 2021b. ISSN 0957-4174. DOI: 10.1016/j.eswa.2021.114784.

4. RIGO, Cezar Antônio; SEMAN, Laio Oriel; CAMPONOGARA, Eduardo; MORSCH FILHO, Edemar; BEZERRA, Eduardo Augusto; MUNARI, Pedro. A branch-and-price algorithm for nanosatellite task scheduling to improve mission quality-of-service. **European Journal of Operational Research**, v. 303, n. 1, p. 168–183, 2022. ISSN 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2022.02.040.

5. FILHO, Edemar Morsch et al. Irradiation Flux Modelling for Thermal–Electrical Simulation of CubeSats: Orbit, Attitude and Radiation Integration. **Energies**, v. 13, n. 24, 2020. ISSN 1996-1073. DOI: 10.3390/en13246691.

6. SEMAN, Laio Oriel; RIBEIRO, Brenda F., et al. An Energy-Aware Task Scheduling for Quality-of-Service Assurance in Constellations of Nanosatellites. **Sensors**, v. 22, n. 10, 2022. ISSN 1424-8220. DOI: 10.3390/s22103715. Available from: https://www.mdpi.com/1424-8220/22/10/3715.

7. CAMPONOGARA, Eduardo et al. A continuous-time formulation for optimal task scheduling and quality-of-service assurance in nanosatellites. **Computers & Operations Research**, v. 147, p. 105945, 2022. ISSN 0305-0548. DOI: https://doi.org/10.1016/j.cor.2022.105945.

2 STATE OF THE ART

This chapter presents the methodology and results of a systematic literature review, performed aiming to identify technological gaps and gather scientific, theoretical, and practical knowledge necessary to situate and substantiate the research proposed in this thesis. Notice that the text of this chapter was partially published as an article in Rigo, Seman, Camponogara, Morsch Filho, and Bezerra (2021).

2.1 METHODOLOGY

The methodology followed here is that of "mapping study" (COOPER, 2016; DIAS; BARBOSA; VIANNA, 2018), developed to minimize researcher bias in the process of finding, filtering and analyzing the published works. This process consists in tree phases:

1. Definition of research questions
2. Design of the search process
3. Definition of criteria for filtering the results

The following three topics will describe in detail all these steps, to make the process of searching and analyzing the literature as transparent as possible.

2.1.1 Research questions

For this work, four General Questions (GQ), three Specific Questions (SQ) and two Publication Questions (PQ) were delineated. The intent with the general questions is to understand what technologies are being employed for modeling and solving the scheduling problem, and what systems are being modeled. The intent with the specific questions is to identify how these techniques are being applied. Lastly, the reason for the publication questions is to find where these studies are being published, and what is their impact. The research questions are presented in Table 1.

2.1.2 Research process

The first step in the research process is to define the search string to be used in the databases search engines. Synonyms, or equivalent expressions, for the main terms “satellite” and “scheduling” were considered, resulting in the following search string: ((satellite OR cubesat OR nanosatellite OR optimal satellite OR satellite energy OR satellite energy-driven) AND (scheduling OR scheduling optimization OR task scheduling OR mission plan)) – see Table 2.

With the search string set, the second step was to specify the publication repositories were to conduct the searches. Taking into account the major repositories in engineering and computer science, seven were chosen: IEEE Xplore Digital Library, ACM Digital Library, arXiv,

Table 1 – Research questions.

Reference	Question
General Questions	
GQ1	What strategies are being used to formulate the task scheduling problem in satellites?
GQ2	What types of missions are being modeled?
GQ3	What types of satellites are being modeled?
GQ4	What algorithms are being used to solve the formulations?
Specific Questions	
SQ1	How are the strategies being applied in the task scheduling problem in satellites?
SQ2	How is optimization being used to address the scheduling problem?
SQ3	How is energy being modeled in these formulations?
Publication Questions	
PQ1	Where have the works been published?
PQ2	What is the average number of citations?

Source: **The author**.

Table 2 – Search terms.

Main term	Synonym
Satellite	satellite OR cubesat OR nanosatellite OR optimal satellite OR satellite energy OR satellite energy-driven
Scheduling	scheduling OR scheduling optimization OR task scheduling OR mission plan

Source: **The author**.

Science Direct, Wiley, Springer Library, and Google Scholar. In order to get more relevant results, the IEEE Xplore search was performed using the *command search* in *other search options* and limiting the search to document title only. On ACM Digital Library, arXiv and Wiley the search text was inserted directly in the search engines. In the Springer Library, on advanced search options, the search query was inserted in the field *where the title contains*. In Science Direct, the advanced search option was used and the search string was entered in the area *Title, abstract or author-specified keywords*. Lastly, for Google Scholar, the search was done using the keyword *allintitle*, which limits the search to titles only.

2.1.3 Publications filtering

For the search results filtering process, both inclusion and exclusion criteria were defined according to Table 3. Applying these criteria allows the exclusion of irrelevant studies while helps to find the most impactful and high-quality researches. Hence, the first step in the filtering process was done per IC2, IC3, EC1, EC2, and EC3 while still in the databases, using the website's tools. The resulting titles were analyzed for the next filtering step, abstract and keywords reading. Following that, the remaining papers were catalogued in a tool called Notion [1] to allow for further organization, labeling, and processing. Figure 3 shows the filtering process steps and the resulting number of papers.

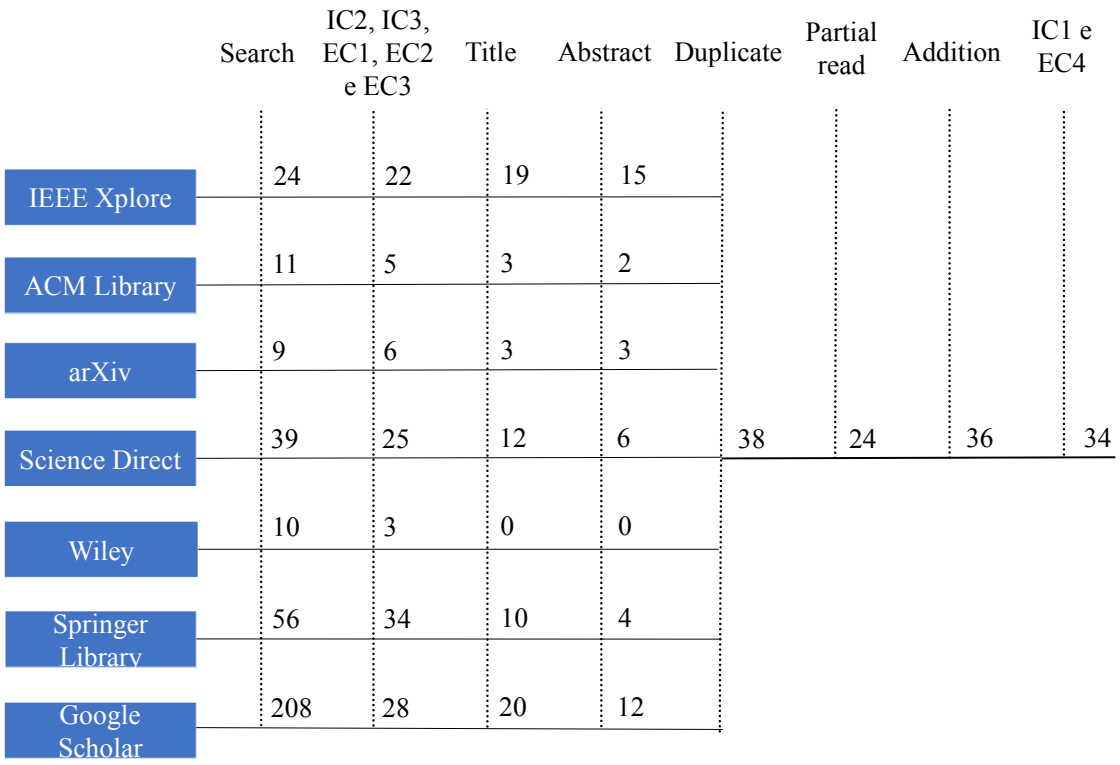
[1] Notion can be accessed in `https://www.notion.so/product/`

Table 3 – Filtering criteria.

Reference	**Criteria**
Inclusion	
IC1	The research brings innovative contributions for formulating and solving the task scheduling problem in satellites.
IC2	The research is published in a journal or a conference.
IC3	The research is a full paper.
Exclusion	
EC1	The work was published before 2010.
EC2	The work was published before 2018 and has no citation.
EC3	The work is not written in English.
EC4	The study is not related to the research questions.

Source: **The author**.

Figure 3 – Filtering process.



Source: **The author**.

After combining all the papers and removing duplicates, a process of partial reading was done for all the remaining. This partial reading followed the guidelines presented by Griswold (2009) and Keshav (2007), on how to read an Engineering paper. The process consisted in carefully reading the abstract, section titles and the conclusion while paying attention to mathematical formulations, figures and tables; all the rest was ignored. This process allowed a basic understanding of the paper contributions, the problem it is formulating, strategies for the solution and its success. With this basic understanding and this thesis research objectives, an

A-to-D relevance classification system was applied according to Hayton (2015), following the criteria below:

- A: The study is crucial for the research, an essential reading.
- B: The study is important and adds to the knowledge of the field, but might not impact this research.
- C: The study may be important and it might be useful.
- D: The study is poorly related.

The papers assigned the letter A and B were fully read, which resulted in finding other relevant studies that were added even if not found by the search process. Therefore, the following papers were added: Fu, Modiano, and Tsitsiklis (2003), Moser et al. (2006), Bianchessi et al. (2007), Edalat and Motani (2016), Hermanns, Krčál, and Nies (2017), Juan Fraire et al. (2018), Nies et al. (2018), Bisgaard et al. (2019), Kørvell and Degn (2019) and Juan A Fraire et al. (2020). The papers assigned the letter C were partially read, aiming to answer the research questions, and the ones with a D were discarded. Finally, the last filtering criterion was to analyze the papers per IC1 and EC4.

As presented in Figure 3, Google Scholar brought the highest number of unrelated studies, when compared with the initial search. Several results from Google Scholar were not from journals or conferences or did not have the full paper written English. Differently, the majority of relevant papers came from IEEE Xplore, which produced the best initial results, and many of them reappeared in Google Scholar search. Wiley had no related papers in their database and ACM Library only 2. By the end of the filtering process, 34 papers remained.

2.2 RESULTS

The selected papers were organized on Notion according to: title, year, class, number of citations, type of mission, type of satellite, scheduling mode (online or offline), scheduling algorithm, constraints and objectives. The papers will be presented in the following sections in order to answer each research question.

2.2.1 GQ1 – What strategies are being used to formulate the task scheduling problem in satellites?

According to Deng et al. (2017), the existing formulations for the satellite scheduling problem fall mainly in three categories: Mathematical Programming Model (MPM), Graph Theory Model (GTM) and Constraint Satisfaction Problem Model (CSPM). Here, however, the MPM category is subdivided to show specifically which researches used MILP for their formulations. The category Others was also created to accommodate works that had unique

approaches, such as Moser et al. (2006) that scheduled based in an Early Deadline First (EDF) policy. Therefore, the papers were classified as presented in Table 4 where the Share column shows the percentage of papers that used each modeling strategy.

Table 4 – Strategies being used to formulate the satellite scheduling problem.

Strategy	**Studies**	**Share**
Mathematical Programming Model	Fu, Modiano, and Tsitsiklis (2003), Juan Fraire et al. (2018), Zhou et al. (2019), Pang et al. (2015), Zhai et al. (2015), Chu, Chen, and Tan (2017), Wang, Demeulemeester, Hu, Qiu, et al. (2018), Cui and Zhang (2019), Wu, Zhang, et al. (2019), Haijiao et al. (2019), He et al. (2019), Lam, Rivest, and Berger (2019) and Juan A Fraire et al. (2020).	39%
Constraint Satisfaction Problem Model	Bianchessi et al. (2007), Wang, Zhu, et al. (2013), Zhu et al. (2014), Wu, Wang, et al. (2014) and Zhao et al. (2019), Deng et al. (2017).	18%
Mixed Integer Linear Programming	Edalat and Motani (2016), Song et al. (2018), Chen et al. (2019), Meng et al. (2019) and Critchley-Marrows, Isacsson, and Gårdebäck (2019).	14%
Graph Theory Model	Wu, Ma, et al. (2012), Wang, Sheng, et al. (2018) and Jia et al. (2017).	9%
Others	Moser et al. (2006), Castro and Straub (2015), Hermanns, Krčál, and Nies (2017), Nies et al. (2018), Slongo et al. (2018), Bisgaard et al. (2019) and Kørvell and Degn (2019).	20%

Source: **The author**.

Some studies utilized combined strategies in their formulation, such as Wu, Ma, et al. (2012) and Wang, Sheng, et al. (2018) that employed both MPM and GTM – although, in the statistics, only one category was accounted for. Others, like Kørvell and Degn (2019), Bisgaard et al. (2019) and Nies et al. (2018) used Priced Timed Automata (PTA) as an alternative technique to classical MPM formulations.

2.2.2 GQ2 – What types of missions are being modeled?

The results showed that the majority of missions are those of Earth Observation (EO), with seventeen papers, seven published in 2019 only. Following that, nine papers modeled missions of data relay satellites, and four papers missions of scientific experiments – such as CubeSats with payloads. One research scheduled a mission for Radio Detection And Ranging (RADAR), and other single research a mission for wireless communications. Two papers included in the addition phase of the filtering process scheduled sensor networks, and not satellite missions. The final classification is shown in Table 5.

2.2.3 GQ3 – What types of satellites are being modeled?

The types of satellites considered here were: single satellite, multiple satellite, and CubeSat. Since CubeSat is a particular type of spacecraft, with its own set of limitations and applications, it was decided to differentiate it from the generic satellite formulations. The works

Table 5 – Type of satellite missions formulated.

Mission	**Researches**	**Share**
Earth observation	Bianchessi et al. (2007), Wu, Ma, et al. (2012), Wang, Zhu, et al. (2013), Zhu et al. (2014), Wu, Wang, et al. (2014), Zhai et al. (2015), Chu, Chen, and Tan (2017), Hermanns, Krčál, and Nies (2017), Wang, Sheng, et al. (2018), Song et al. (2018), Wang, Demeulemeester, Hu, Qiu, et al. (2018), Cui and Zhang (2019), Wu, Zhang, et al. (2019), Haijiao et al. (2019), He et al. (2019), Chen et al. (2019) and Lam, Rivest, and Berger (2019).	50%
Data relay satellite	Deng et al. (2017), Jia et al. (2017), Nies et al. (2018), Juan Fraire et al. (2018), Kørvell and Degn (2019), Bisgaard et al. (2019), Zhao et al. (2019), Zhou et al. (2019) and Juan A Fraire et al. (2020).	26%
Scientific experiments	Castro and Straub (2015), Slongo et al. (2018), Meng et al. (2019) and Critchley-Marrows, Isacsson, and Gårdebäck (2019).	12%
Sensor network	Moser et al. (2006) and Edalat and Motani (2016).	5%
Wireless communication	Fu, Modiano, and Tsitsiklis (2003).	3%
RADAR	Pang et al. (2015).	3%

Source: **The author**.

classification is presented in Table 6 and the 7% missing in the Share column accounts for the two papers on sensor networks – see Table 5.

Table 6 – Type of satellite modeled.

Satellite	**Researches**	**Share**
Multiple	Bianchessi et al. (2007), Wu, Ma, et al. (2012), Zhu et al. (2014), Wu, Wang, et al. (2014), Zhai et al. (2015), Deng et al. (2017), Jia et al. (2017), Wang, Demeulemeester, Hu, Qiu, et al. (2018), Cui and Zhang (2019), He et al. (2019), Chen et al. (2019), Meng et al. (2019) and Lam, Rivest, and Berger (2019).	38%
Single	Fu, Modiano, and Tsitsiklis (2003), Wang, Zhu, et al. (2013), Chu, Chen, and Tan (2017), Song et al. (2018), Wu, Zhang, et al. (2019), Zhao et al. (2019), Haijiao et al. (2019) and Zhou et al. (2019).	23%
Single CubeSat	Bisgaard et al. (2019), Nies et al. (2018), Hermanns, Krčál, and Nies (2017), Pang et al. (2015), Castro and Straub (2015), Slongo et al. (2018) and Critchley-Marrows, Isacsson, and Gårdebäck (2019).	20%
Multiple CubeSats	Juan Fraire et al. (2018), Juan A Fraire et al. (2020) and Kørvell and Degn (2019)	9%

Source: **The author**.

2.2.4 GQ4 – What algorithms are being used to solve the formulations?

The algorithms being used for solving the formulations in task scheduling vary greatly, from commercial solvers to unique specific approaches. Wang, Demeulemeester, Hu, Qiu, et al. (2018) identifies four major groups:

- Exact algorithms: includes approaches that can guarantee optimal solutions, such as branch-and-bound, Russian Doll search, dynamic programming and commercial solvers;
- Meta-heuristics: includes approaches that use stochastic search, such as tabu search, ant colony, evolution algorithms, local search and simulated annealing;
- Heuristics: includes problem specific approaches based on problem-specific rules, such as EDF or greedy and constructive algorithms.
- Priced Timed Automata: a methodology that extends well-established timed automata technology to optimal scheduling and planning problems (BEHRMANN; LARSEN; RASMUSSEN, 2005).

Here, one more categorie will be considered, Deep Learning. Therefore, the classification of researches is presented in Table 7. The majority of works utilizes more than one approach in the resolution, so many papers will appear in more than one category.

Table 7 – Algorithms being used to solve the scheduling formulations.

Algorithm	**Researches**
Heuristic	Fu, Modiano, and Tsitsiklis (2003), Moser et al. (2006), Wang, Zhu, et al. (2013), Zhu et al. (2014), Wu, Wang, et al. (2014), Zhai et al. (2015), Edalat and Motani (2016), Jia et al. (2017), Wang, Sheng, et al. (2018), Song et al. (2018), Wang, Demeulemeester, Hu, Qiu, et al. (2018), Wu, Zhang, et al. (2019), He et al. (2019), Meng et al. (2019) and Lam, Rivest, and Berger (2019).
Exact	Juan Fraire et al. (2018), Juan A Fraire et al. (2020), Hermanns, Krčál, and Nies (2017), Fu, Modiano, and Tsitsiklis (2003), Bianchessi et al. (2007), Pang et al. (2015), Chu, Chen, and Tan (2017), Wang, Demeulemeester, Hu, Qiu, et al. (2018), Critchley-Marrows, Isacsson, and Gårdebäck (2019), Chen et al. (2019), Zhou et al. (2019) and Haijiao et al. (2019).
Meta-heuristics	Bianchessi et al. (2007), Wu, Ma, et al. (2012), Zhai et al. (2015), Deng et al. (2017), Cui and Zhang (2019), Wu, Zhang, et al. (2019), Zhao et al. (2019) and He et al. (2019).
Deep Learning	Haijiao et al. (2019), Meng et al. (2019) and Lam, Rivest, and Berger (2019).
Priced Timed Automata	Bisgaard et al. (2019), Nies et al. (2018) and Kørvell and Degn (2019).

Source: **The author**.

2.2.5 SQ1 – How are the strategies being applied in the task scheduling problem in satellites?

The majority of works apply the strategies to solve the problem online and in a non-optimal form. The online solutions are real-time or near real-time for scheduling onboard the spacecraft. The offline solutions are usually solved on the ground and are sent to the satellite via telecommand. Table 8 shows which papers reported online and offline scheduling modes.

Table 8 – Scheduling modes.

Mode	Researches	Share
Online	Moser et al. (2006), Wang, Zhu, et al. (2013), Zhu et al. (2014), Wu, Wang, et al. (2014), Zhai et al. (2015), Edalat and Motani (2016), Deng et al. (2017), Chu, Chen, and Tan (2017), Jia et al. (2017), Slongo et al. (2018), Wang, Sheng, et al. (2018), Song et al. (2018), Cui and Zhang (2019), Wu, Zhang, et al. (2019), Meng et al. (2019), Zhou et al. (2019), Haijiao et al. (2019) and Lam, Rivest, and Berger (2019).	53%
Offline	Nies et al. (2018), Kørvell and Degn (2019), Juan Fraire et al. (2018), Juan A Fraire et al. (2020), Hermanns, Krčál, and Nies (2017), Bisgaard et al. (2019), Fu, Modiano, and Tsitsiklis (2003), Bianchessi et al. (2007), Pang et al. (2015), Wang, Demeulemeester, Hu, Qiu, et al. (2018), Chen et al. (2019) and Critchley-Marrows, Isacsson, and Gårdebäck (2019).	35%

Source: **The author**.

Wu, Ma, et al. (2012) and Zhou et al. (2019) were not clear in their paper about the application of their algorithm; therefore, they had no classification. He et al. (2019) provided both solutions, one offline and one online. The only studies to provide an optimal schedule were those of Moser et al. (2006) and Chen et al. (2019).

2.2.6 SQ2 – How is optimization being used to address the scheduling problem?

Optimization usually requires an objective function and a set of constraints to be defined. In scheduling applications the objective functions are formulated to maximize mission profit (WANG; DEMEULEMEESTER; HU; QIU, et al., 2018; WU; ZHANG, et al., 2019), maximize sum of task priorities (MENG et al., 2019; DENG et al., 2017), maximize data throughput (ZHOU et al., 2019; FRAIRE, J. et al., 2018; FRAIRE, J. A. et al., 2020; KØRVELL; DEGN, 2019), maximize image resolution in EO missions (ZHU et al., 2014), minimize resource utilization (WANG; ZHU, et al., 2013; ZHAO et al., 2019), and other very specific goals such as to minimize the overlap between satellites (HE et al., 2019).

The type of constraints varies greatly. EO missions, for example, usually include constraints for memory capacity, attitude control, position, orbit and timing (HAIJIAO et al., 2019; CHEN et al., 2019; CUI; ZHANG, 2019). Data relay satellite scheduling formulations usually include constraints of memory, bandwidth, position, energy, and attitude (ZHAO et al., 2019; ZHOU et al., 2019; DENG et al., 2017).

Finally, the results showed that all of the papers which presented offline solutions employed mathematical optimization formulations.

2.2.7 SQ3 – How is energy being modeled in these formulations?

As shown in Table 9, the majority of works formulated energy as a constraint on their MPM. Some works considered it in their objective functions, such as Zhao et al. (2019), Deng et al. (2017), Pang et al. (2015) and Wang, Zhu, et al. (2013) that modeled it for minimization of power consumption and Edalat and Motani (2016) which aimed to balance energy levels across the sensor nodes. Slongo et al. (2018) and Moser et al. (2006), who did not employ MPM, considered it as a resource to be fully harvested and a limiting factor, respectively. Finally, 35% of the researches did not account for energy resources at all.

Table 9 – Energy in the satellite scheduling formulations.

Formulated as	**Researches**	**Share**
Constraint	Fu, Modiano, and Tsitsiklis (2003), Wu, Ma, et al. (2012), Wu, Wang, et al. (2014), Wang, Demeulemeester, Hu, Qiu, et al. (2018), Song et al. (2018), Cui and Zhang (2019), Wu, Zhang, et al. (2019), Zhou et al. (2019), He et al. (2019), Lam, Rivest, and Berger (2019), Critchley-Marrows, Isacsson, and Gårdebäck (2019) and Kørvell and Degn (2019).	35%
Not accounted	Bianchessi et al. (2007), Zhu et al. (2014), Castro and Straub (2015), Zhai et al. (2015), Jia et al. (2017), Chu, Chen, and Tan (2017), Wang, Sheng, et al. (2018), Haijiao et al. (2019), Chen et al. (2019) and Meng et al. (2019).	29%
Objective	Juan A Fraire et al. (2020), Juan Fraire et al. (2018), Pang et al. (2015), Wang, Zhu, et al. (2013), Edalat and Motani (2016), Deng et al. (2017) and Zhao et al. (2019).	20%
Other	Moser et al. (2006), Slongo et al. (2018), Bisgaard et al. (2019), Nies et al. (2018) and Hermanns, Krčál, and Nies (2017).	15%

Source: **The author**.

2.2.8 PQ1 – Where have the works been published?

The majority of papers have been published in high impact journals, such as Acta Astronautica, IEEE Systems Journal, IEEE Access and IEEE Transactions on Parallel and Distributed Systems. Only five papers have been published in conferences: Moser et al. (2006), Zhao et al. (2019), Castro and Straub (2015), Meng et al. (2019) and Lam, Rivest, and Berger (2019).

2.2.9 PQ2 – What is the average number of citations?

The average number of citations for the 34 resulting papers was 22.67. The most cited one is Bianchessi et al. (2007) with 189 citations.

2.3 DISCUSSION

In this section, the most relevant researches will be briefly discussed, the majority of which were previously classified as A.

Earth observation missions are modeled in several works. Bianchessi et al. (2007), for instance, tackled the problem of scheduling earth observation requests in a scenario comprised of several satellites and multiple orbits. Implementing an algorithm based on tabu search heuristics, their objective was to maximize profit determining which requests to attend while conforming to operational constraints of orbit, position and time. A column generation technique was used to set upper bounds and assess the quality of the solutions, which were not optimal. The similar work of Wang, Demeulemeester, Hu, Qiu, et al. (2018), formulated a stochastic model of clouds and a novel mixed-integer nonlinear programming problem to schedule earth observation requests in a scenario with multiple satellites. They developed two column-based heuristics and a knapsack-based heuristic to find feasible solutions for large-scale instances. Zhu et al. (2014) also explored the multiple satellite scheduling problem for earth observation, introducing fault-tolerance mechanisms in a double-objective real-time scheduling algorithm. The authors introduced the fault-tolerance capabilities by implementing techniques of primary-backup with task overlapping, task merging, and task insertion in multiple satellites, enhancing reliability and observation resolutions.

Other commonly addressed problem is emergency response. Wu, Zhang, et al. (2019) compare scheduling algorithms and propose a novel task priority model for this problem. They formulate a dual-objective optimization problem for scheduling and solve it using heuristic, genetic and NSGA-II[2] algorithms for comparison. Their results showed that NSGA-II algorithm is the most effective in finding the best objective values while the heuristic algorithm has less computational cost. The analogous work of Wang, Zhu, et al. (2013) established constraints for backward shift of tasks to allow more flexibility in the insertion of new tasks and for task merging to execute multiple tasks simultaneously. Combined to that, the authors formulated a mathematical programming model with multiple objectives and employing heuristics, were able to solve the problem as to outperform alternatives in their results. Cui and Zhang (2019) also explored dynamic real-time scheduling for emergency response in earth observation satellites. To solve the scheduling problem, they created a hybrid genetic tabu search algorithm. They also presented an algorithm for dynamic insertion, reallocation and deletion of tasks to the initial scheduling plan based on parameters such as task urgency, type of observation, conflicts, and execution viability.

Communications satellites are also a topic of intense research. Fu, Modiano, and Tsitsiklis (2003), for instance, researched energy consumption for communication satellites presenting a dynamic programming formulation for maximizing users' requests response constrained to the amount of available energy. Nevertheless, in their formulation, if the battery is full and

[2] NSGA-II is a stochastic evolutionary solving algorithm for multi-objective optimization formulations. More details in: `https://ieeexplore.ieee.org/document/996017`

the users request does not consume all energy input from solar cells, the energy is lost. In contrast, Deng et al. (2017) proposes two methodologies, one for static and other for dynamic task scheduling in data relay satellite systems. A multi-objective optimization formulation was created by the authors to maximize the sum of task priorities and minimize power consumption. They formulated constraints for the satellite's antenna mechanisms (switching time, data rate, number of users), scheduling delay, and memory usage, finding an initial scheduling solution applying a genetic algorithm. In the dynamic task model, tasks can be preempted, divided, time dislocated, and switched, although the objectives here are to minimize task divisions and overall changes while maximizing their sum weight.

Nanosatellite missions are also studied. Both Bisgaard et al. (2019) and Nies et al. (2018) report on the use of Priced Timed Automata (PTA) for battery-aware scheduling a GOMX–3 mission. The PTA technique is a new and promising technique for task scheduling, nevertheless is not as rigorous as other approaches such MP, which allows for flexibility, that was very well exploited by the authors, but can result in important jobs not executing. Although they successfully demonstrated the application of the technique, their solution is tailored for the GOMX–3 mission and their energy model considers constant solar energy on non-eclipse time, a non-ideal approximation. Furthermore, despite reaching optimality, their objective function is cost-optimal and not reward-optimal and does not maximize mission value. Similarly, Critchley-Marrows, Isacsson, and Gårdebäck (2019) describes the process of formulating a mixed-integer linear programming model to schedule eight payloads in a 2U Swedish CubeSat mission, constrained to power availability and specific requirements for each payload. Their objective was to obtain a feasible schedule and did not account for payloads priority or maximization of mission value. Although they succeeded in scheduling the payloads for several orbits, their basic formulation was too mission-specific, lacking generality for easy reusability. Slongo et al. (2018) also focused on CubeSats, presenting a scheduling algorithm based in the perturb and observe strategy to maintain the solar panels in their maximum power point by dynamically activating tasks of fitting power consumption, in order to match power input. Although their goal was similar to the one proposed here and they have empirically demonstrated that task execution can maximize energy harvesting, their algorithm neither accounts for quality of service nor does it provide an optimal task plan. Moreover, lists of tasks were required to be manually organized, therefore not fully automating the scheduling process.

Pang et al. (2015) is another relevant work that discussed the application of nanosatellites in swarm for synthetic aperture radar (SAR) imaging. The authors present a mathematical formulation for job scheduling based on constraints of stochastic failures probability and communication bandwidth with the objective of minimizing power consumption and improving reliability. A convexification method was used to reduce the computational demand and a cutting-plane algorithm was employed to solve the problem. Their algorithm showed improvements in metrics of mission failure when compared to general scheduling algorithms.

Previous research has also addressed battery models for energy management in satellites.

In Hermanns, Krčál, and Nies (2017), the authors develop and explore a comprehensive kinetic battery model to compute the risk of premature battery depletion given uncertainty in the load process and a constant charging behavior. Using a Markov decision process, they developed a task scheduling algorithm and applied to a real 2U CubeSat mission (GOMX-1). Despite demonstrating a successful schedule for the mission, their focus is on battery lifetime analysis, probability of depletion and state of charge in long time-spans. Consequently, their scheduling approach is not focused on QoS guarantees, lacks generality for easy re-usability and does not find optimal solutions. Furthermore, their energy model considers constant solar energy on non-eclipse time.

2.4 CONCLUSIONS

From the prior survey, some conclusions can be drawn. The research question's answers showed that Mathematical Programming (MPM) is the most used strategy to formulate the satellite scheduling problem, given that 57% of the studies used it. These formulations are primarily being solved with heuristics or meta-heuristics algorithms for online scheduling in a non-optimal manner.

The majority of formulations are of earth observation missions with multiple satellites, where real-time scheduling is of interest for user requests response and emergency response. Moreover, from the selected papers published in the period 2017-2019, 65% of them are for EO scheduling, characterizing it as a recent research trend. Another possible trend is the implementation of Deep Learning techniques, given that three 2019 publications used it in an attempt to automate online decision making about which tasks to execute next.

On the other hand, little research has been carried on the problem of scheduling single nanosatellite missions, even though this type of spacecraft has great potential and is playing an increasing role in the space sector. They are a type of satellite with its own set of limitations and applications, and seven studies on scheduling modeling have been found in this survey, all of them with limited approaches and results.

Furthermore, little research has been dedicated in modeling and managing energy resources. Even though it is a crucial factor in satellite systems, 29% of the researches did not account for energy in any form on their formulations. The majority of papers considered it as a simple constraint, or a resource to be minimally exploited – portrayed by objective functions to minimize energy consumption. The researches that best analyzed the energy problem, considering battery and a harvesting unit, were those of Edalat and Motani (2016) and Moser et al. (2006); nevertheless, they are focused on sensor networks in the ground weather and not satellites in orbit.

However, no study contemplates single nanosatellite under variant power input with the rigor and advantages of exact optimization algorithms. Thus, the work presented in this thesis addresses these two combined features, namely the analysis and modeling of the nanosatellite scheduling problem aiming at maximizing energy resource usage, and the optimization of the

mission extraction of value.

3 BASE MODEL FORMULATION

This chapter presents the methodology for task scheduling, optimal power management and QoS assurance. The text here was published as an article in Rigo, Seman, Camponogara, Morsch Filho, and Bezerra (2021).

First the satellite power input model used is detailed, in Section 3.1, based on the methodology described by Filho et al. (2020). Notice that the power input methodology was not developed, nor was it implemented by the author, and is only presented in this thesis so the reader can fully understand the work. The calculated orbit power is then used as an input vector for the scheduler. Then, the novel optimal scheduling formulation, which is cast in the form of a mathematical programming problem, is presented in Section 3.2. This chapter also presents the simulation scenarios explored and the results and analysis, in Section 3.3; lastly, a brief discussion is presented in Section 3.4.

3.1 POWER INPUT MODEL

To calculate the irradiance over the CubeSat, and consequently the power input, an analytical model was used. Encompassing different parameters of orbit and attitude, the model uses a rotation matrix to simulate the dynamics of the satellite along the orbit (FILHO et al., 2020). This model can be extended for bigger and different geometries by a simple adaptation of normal vectors that represent the body. It does not have limitations in terms of spin speed neither spin axis, it is, however, suitable for a perfectly circular orbit without the typical lag in the ascending node. Here a time discretization is applied, such that the power harvested will be considered constant in a given time period, usually of one minute length. This is a reasonable approximation given the kinematics of the spacecraft is relatively constant in such small time window, and consequently the power harvested.

Radiation from the Sun is the most significant energy source for satellites in Low Earth Orbit (LEO). The constant value adopted in this study is $I_{sun} = 1360 W/m^2$ for the solar flux radiation and it is assumed that the solar rays are parallel. However, the position and orientation of the satellites may be transient and both orbital parameters and attitude will define the amount of surface-reaching irradiance, duration of the eclipse, which sides are shadowed by nearby neighborhood and consequently the total power input. The power generation by photo-voltaic panels (P_k) on each one of the six sides k of a standard CubeSat is modeled by:

$$P_k = \eta A_{pv_k} I_{sun} F_{k \to Sun} \Psi \tag{1}$$

where η is the efficiency of the photo-voltaic cells in the emission spectrum of the Sun, A_{pv_k} is the area of the photovoltaic cells, $F_{k \to Sun}$ is the view factor of the surface k in relation to the Sun and finally the parameter Ψ is a step function created to become null when the satellite is under the shadow of the Earth. Further details about the development of these equations can be found in EdemarIrradiance.

Assuming Commercial Off The Shelf (COTS) model for the CubeSat's solar panel, the efficiency and area of photovoltaic cells are 30% and 60.36 cm^2 for 1U (GOMSPACE, 2020), respectively. For 2U, in four sides the area duplicates, while the efficiency remains the same for every side, as in the 1U.

3.2 OPTIMAL SCHEDULING ALGORITHM

Before presenting the mathematical formulation of the proposed optimal scheduling algorithm, the definitions of variables and sets are presented in Table 10, and the parameters in Table 11. Given a set of tasks $\mathcal{J}$ and a set of time units $\mathcal{T}$, the scheduling algorithm must allocate each task (or job) $j \in \mathcal{J}$ to a subset of time units $t \in \mathcal{T}$. Note that time is discretized here, usually in a minute by minute granularity, which is a reasonable simplification given that tasks are hardware modules and do not require finer control over start and finish execution time.[1] This allocation must be done for each of the satellite subsystems $s \in \mathcal{S}$ whilst meeting all requirements for every task.

Table 10 – Sets, indices and decision variables.

Notation	**Definition**
Sets	
$\mathcal{S} = \{1, \ldots, S\}$	set of satellite subsystems.
$\mathcal{J} = \{1, \ldots, J\}$	set of tasks to be scheduled.
$\mathcal{T} = \{1, \ldots, T\}$	set of time periods.
Variables	
$x_{sjt} \in \{0, 1\}$	takes the value of 1 if, and only if, task $j \in \mathcal{J}$ on subsystem $s \in \mathcal{S}$ is in execution at time $t \in \mathcal{T}$.
$\phi_{sjt} \in \{0, 1\}$	takes the value of 1 if, and only if, the execution of task $j \in \mathcal{J}$ on subsystem $s \in \mathcal{S}$ initiates at time $t \in \mathcal{T}$.

Source: **The author**.

Usually, satellite tasks are periodic, requiring to be executed each $p_{sj}^{\min}$ units of time (minimum period). Likewise, in addition to consuming time, tasks have a power footprint, requiring q_{sj} amount of energy to be executed. Therefore, the sum of power consumption by all jobs at any moment t cannot be greater than r_t. Another important characteristic, for quality of service guarantee, is to perform the scheduling based on each job priority u_{sj}. Consequently, the integer programming model for satellite task scheduling was formulated as follows.

Considering all jobs and time units for each subsystem, the task scheduling problem is formulated to maximize function (2), which is the sum of task priority times the activation variable ($u_{sj}x_{sjt}$). This ensures that tasks with higher priority will induce a larger objective value

[1] A continuous time methodology has been explored by the author in the work presented by Camponogara et al. (2022), and the results showed such approach can solve representative problem instances in a considerably shorter time, while being able to deal with real-valued constraint functions.

Table 11 – Parameters.

Notation	Definition
S	number of subsystems.
J	number of tasks.
T	number of time periods.
r_t	amount of power input at time t (from solar panels).
q_{sj}	power consumed by executing a task j on subsystem s.
u_{sj}	priority of task j on subsystem s.
$t_{sj}^{\min}/t_{sj}^{\max}$	minimum/maximum CPU time of task j on subsystem s.
$y_{sj}^{\min}/y_{sj}^{\max}$	minimum/maximum number of startups of task j on subsystem s.
$p_{sj}^{\min}/p_{sj}^{\max}$	minimum/maximum period of task j on subsystem s.
$w_{sj}^{\min}/w_{sj}^{\max}$	start/finish moment for the execution window of task j on subsystem s.

Source: **The author**.

and, therefore, preference in time allocation.

$$F: \max_{x_{sjt}} \sum_{s=1}^{S}\sum_{j=1}^{J}\sum_{t=1}^{T} u_{sj}x_{sjt} \tag{2}$$

Constraints (3) state that the power used by an active job ($q_{sj}x_{sjt}$) must be less than or equal to the maximum available energy resource at each moment (r_t). Therefore, this is the energy constraint, which ensures that the task total power consumption stays below the available amount for every moment along the course of an orbit.

$$\sum_{s=1}^{S}\sum_{j=1}^{J} q_{sj}x_{sjt} \leq r_t, \qquad \forall t \in \mathcal{T}, \tag{3}$$

Constraints (4a) to (4d) ensure that the auxiliary variable ϕ detects the startup of a job by taking the value of 1 only at the unit of time that a job is initiated. Constraints (4b) and (4c) ensure ϕ takes the value of 1 whenever a task is initiated – the moment x goes from zero to one. Constraints (4b) and (4d) ensure that ϕ returns to the value of zero after assuming the value one – even if x continues to assume the value 1. Finally, constraints (4a) ensure the correct value of ϕ for $t = 1$.

$$\phi_{sj1} \geq x_{sj1}, \qquad \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \tag{4a}$$

$$\phi_{sjt} \geq x_{sjt} - x_{sj(t-1)}, \qquad \forall j \in \mathcal{J}, \forall t > 1, \forall s \in \mathcal{S}, \tag{4b}$$

$$\phi_{sjt} \leq x_{sjt}, \qquad \forall j \in \mathcal{J},\ \forall t \in \mathcal{T}, \forall s \in \mathcal{S}, \tag{4c}$$

$$\phi_{sjt} \leq 2 - x_{sjt} - x_{sj(t-1)}, \qquad \forall j \in \mathcal{J}, \forall t > 1, \forall s \in \mathcal{S}, \tag{4d}$$

With the assurance that ϕ will assume the value one every time its associated job is executed, the constraints (5a) and (5b) were introduced to ensure that the sum of all ϕ_t along an

orbit stays within the minimum and the maximum number of startups required for the associated job.

$$\sum_{t=1}^{T} \phi_{sjt} \geq y_{sj}^{\min}, \qquad \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \tag{5a}$$

$$\sum_{t=1}^{T} \phi_{sjt} \leq y_{sj}^{\max}, \qquad \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \tag{5b}$$

The constraints (6a), (6b) and (6c) ensure that the time allocated for the execution of each job j (when flagged by ϕ_{sjt}) is within the minimum and the maximum Central Processing Unit (CPU) execution time required by it. Since x takes the value of one for each unit of time the associated job is in execution, constraint (6a) was formulated so that in a time window of size $t^{\min}$ the sum of the x values is greater than or equal to $t^{\min}$. The multiplication by ϕ_{sjt} on the right-hand side is necessary to enforce the minimum execution time only if job j starts at the time t, as flagged by $\phi_{sjt} = 1$. Likewise, constraints (6b) follow the same logic, although for $t^{\max}$.

$$\sum_{l=t}^{t+t_{sj}^{\min}-1} x_{sjl} \geq t_{sj}^{\min} \phi_{sjt}, \qquad \forall t \in \{1, \ldots, T - t_{sj}^{\min} + 1\}, \ \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \tag{6a}$$

$$\sum_{l=t}^{t+t_{sj}^{\max}} x_{sjl} \leq t_{sj}^{\max}, \ \forall t \in \{1, \ldots, T - t_{sj}^{\max}\}, \qquad \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \tag{6b}$$

$$\sum_{l=t}^{T} x_{sjl} \geq (T - t + 1)\phi_{sjt}, \qquad \forall t \in \{T - t_{sj}^{\min} + 2, \ldots, T\}, \ \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \tag{6c}$$

Given that the previous constraints are valid only for values of t from one to $T - t^{\min}$ and one to $T - t^{\max}$, constraints (6c) were created to ensure that if a task starts execution after these periods, it remains in execution until the end of the orbit.

Constraints (7a) were formulated to ensure that a task is not invoked more than once in any time window of size $p^{\min}$. Similarly, constraints (7b) ensure the task is executed at least once in a given period of size $p^{\max}$.

$$\sum_{l=t}^{t+p_{sj}^{\min}-1} \phi_{sjl} \leq 1, \qquad \forall t \in \{1, \ldots, T - p_{sj}^{\min} + 1\}, \ \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \tag{7a}$$

$$\sum_{l=t}^{t+p_{sj}^{\max}-1} \phi_{sjl} \geq 1, \qquad \forall t \in \{1, \ldots, T - p_{sj}^{\max} + 1\}, \ \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \tag{7b}$$

Constraints (8a) were created to ensure that x takes the value of zero for every unit of time before the moment $w^{\min}$, ensuring the job will not execute in this period. Likewise, constraints (8b) prevent x from taking the value of one from the moment $w^{\max}$ until the end of the orbit.

$$x_{sjt} = 0, \qquad \forall t \in \{1, \ldots, w_{sj}^{\min}\}, \ \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \tag{8a}$$

$$x_{sjt} = 0, \qquad \forall t \in \{w_{sj}^{\max}, \ldots, T\},\ \forall j \in \mathcal{J}, \forall s \in \mathcal{S}, \tag{8b}$$

Finally, (9a) and (9b) establish ϕ and x as binary variables.

$$x_{sjt} \in \{0,1\}, \qquad \forall j \in \mathcal{J},\ \forall t \in \mathcal{T}, \forall s \in \mathcal{S}, \tag{9a}$$

$$\phi_{sjt} \in \{0,1\}, \qquad \forall j \in \mathcal{J},\ \forall t \in \mathcal{T}, \forall s \in \mathcal{S}, \tag{9b}$$

3.2.1 Solving the optimization problem

The problem presented in this section is an IP problem, and as such can be cast in the following canonical form:

$$\max\ c^T \mathbf{x} \tag{10a}$$

$$\text{s.t. } A\mathbf{x} \leq b \tag{10b}$$

$$\mathbf{x} \geq 0, \tag{10c}$$

$$\mathbf{x} \in \mathbb{Z}^n, \tag{10d}$$

where $\mathbf{x}$ is a vector defined by stacking integer variables (the decision variables to the optimized); the parameter vector b and the matrix A are easily obtained from the coefficients of the previously presented formulations. Notice that the problem is classical in the optimization literature, and the process to convert any linear problem into their corresponding standard form is straightforward, resulting in no loss of generality.

Integer problems given by (10) can be solved for the global optimum by the use of algorithms such as the branch-and-bound method (VANDERBEI, 2001). The algorithm is available in commercial and non-commercial solvers, like Gurobi (GUROBI OPTIMIZATION, 2016) and CBC (FORREST et al., 2018).

3.3 RESULTS AND ANALYSIS

The applicability of the methodology presented heretofore is demonstrated here by scheduling three nanosatellite missions, each consisting of a different size: a 1U, a 2U, and a 3U. The 1U CubeSat considered in this study is comprised by six Printed Circuit Board (PCB), three essentials for core functionalities – an On-Board Data Handler (OBDH), a Telemetry, Tracking and Command (TT&C) module and the Electrical Power System (EPS) – and three payloads. Having twice and three times more room, but with the same number of core modules, 2U and 3U nanosatellites may accommodate significant more payloads as they can capture more energy, given the surface area increase of the solar panels. Hence, the scheduling demonstration presented here is accomplished by determining at which moments these payloads can be activated, in a minute by minute basis, for a full orbit, in order to maximize the modules' operation time given the available power and operational parameters.

3.3.1 Power input estimation

The resulting power input, computed minute by minute, represents the available power at the solar panel outputs. Consequently, to obtain the energy available to payload usage, these values were multiplied by the EPS efficiency (0.85), and the power consumption of the core modules was subtracted (0.3 W). The consumption and efficiency figures are those of the FloripaSat-I mission, a 1U CubeSat developed by students at the Federal University of Santa Catarina (UFSC) and launched in 2019 (MARCELINO et al., 2020). The same data regarding efficiency and power consumption in the core modules was used in the 2U and 3U examples. Figures 4, 5 and 6 show the resulting available power for each scenario.

The peak of power input among the scenarios were compared, and for both Nadir and Magnet attitude in eclipse, the ratio of peak power input for a 2U Cubesat is 1.73 x 1U, and for a 3U Cubesat is 2.50 x 1U. On the other hand, considering the same attitudes but out of eclipse, the ratio of peak power input for a 2U Cubesat is 2.00 x 1U, and for a 3U Cubesat is 3.00 x 1U. The explanation for this relies on the fact that the standard 2U and 3U always have two sides (top and bottom) that do not grow according to the laterals of the satellite. When these sides are exposed to the Sun under some angle, their contribution to the total power input are limited by their sizes.

3.3.2 Optimization results

The formulation presented in Subsection 3.2 was implemented in the Julia programming language using the JuMP library (DUNNING; HUCHETTE; LUBIN, 2017) and solved using the Gurobi solver (GUROBI OPTIMIZATION, 2016) in a Personal Computer (PC) with Intel(R) Core(TM) i7-8550U 1.8 GHz, 16GB of Random Access Memory (RAM) and Windows 10 64 bits. The algorithm input data and results are presented and discussed in the following sections.

3.3.2.1 1U Nanosatellite

The data used to schedule the 1U payloads are presented in Table 12. The majority of 1U nanosatellite missions are characterized by payloads for educational purposes and/or scientific experiments, where it is of interest to maximize operation time and data acquisition. To that end, for this scenario, the maximum CPU time and maximum period were set to the maximum possible value so that constraints (6b) and (7b) do not limit the payloads operation time. Likewise, the parameters for execution window were set not to restrict execution. Figure 4 shows the resulted scheduling and its respective power footprint in comparison to the available power. Notice that tasks are labeled from A to C, the x-axis spans from time 0 to time T, and the y-axis indicates if the task is on or off for each unit of time. Computation time for Orbit 1 (with eclipse) was 150 milliseconds and for Orbit 2 (without eclipse) 230 milliseconds.

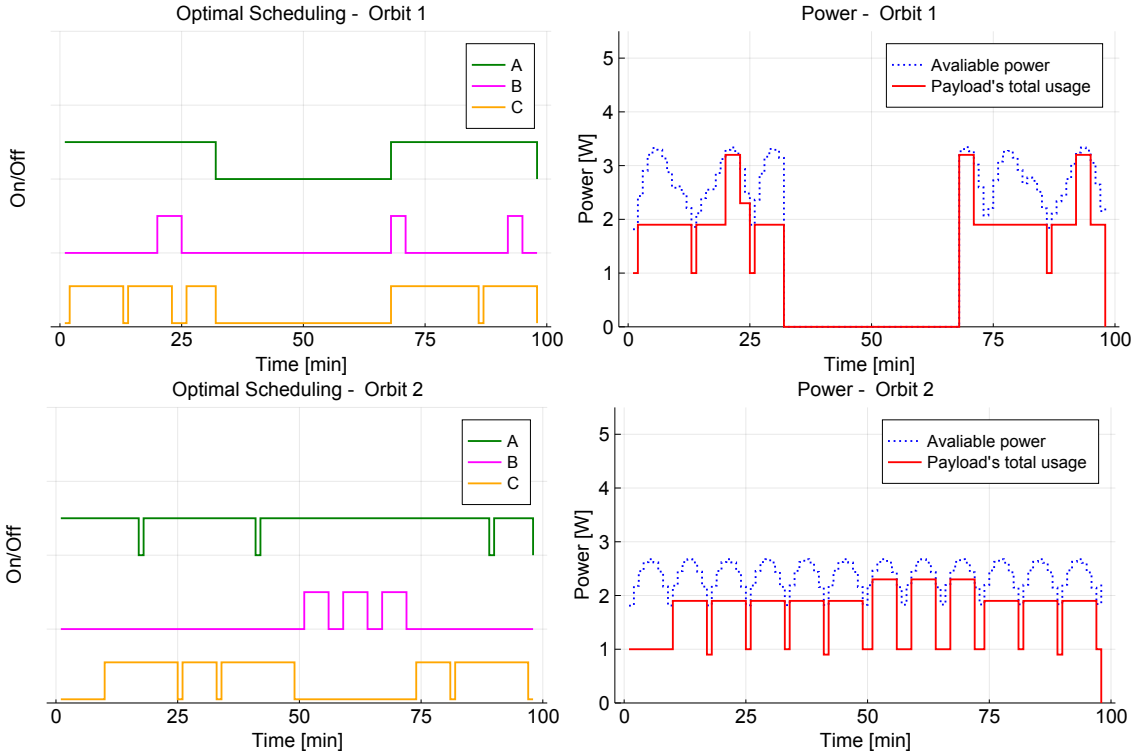
Although it is not a complex example, the resulting optimal scheduling plan for the “1U Orbit 1” scenario shows that task parameters are rigorously followed. Payload A has the highest

Table 12 – 1U payloads scheduling data.

Payload	j	u_j	q_j[w]	$y_j^{\min}$	$y_j^{\max}$	$t_j^{\min}$	$t_j^{\max}$	$p_j^{\min}$	$p_j^{\max}$	$w_j^{\min}$	$w_j^{\max}$
A	1	2	1	1	4	10	100	2	100	0	100
B	2	1	1.3	3	3	3	100	2	100	0	100
C	3	1	0.9	2	5	4	100	3	100	0	100

Source: **The author**.

Figure 4 – 1U simulation results



Source: **The author**.

priority and, therefore, it is only turned off during the eclipse period. In contrast, Payloads B and C, with lower priority, are turned off every time the power is not sufficient to maintain their operation. In consequence, they are reactivated whenever possible, although still limited by their maximum activation parameter $y_j^{\max}$ (3 and 5, respectively). In the resulting optimal scheduling plan for the "Orbit 2" scenario, the available power is never sufficient to maintain both payloads B and C on simultaneously. It can be seen that C is deactivated so that B can be reactivated and fulfill its minimum activation number. The reason for that relies on the fact that the combination orbit and attitude in Orbit 2 exposes four sides of the CubeSat to the Sun, while in the "Orbit 1" scenario all of the six sides are exposed.

Despite the optimal resources utilization given by the resulting scheduling plan, it can be seen that the task power consumption does not perfectly match the available power. In the case of a typical CubeSat, the remaining unused energy can still be harvested and stored in the satellite's battery for usage during eclipse. In the "Orbit 1" scheduling, for instance, 27.09% of

the available power was not immediately utilized by the payloads, which could allow for a steady consumption of 1.27 W during eclipse – if fully captured by the battery.

3.3.2.2 2U Nanosatellite

For the illustrative scenario of a 2U nanosatellite, it is assumed that a second cube unit can allocate 8 payloads in total. Therefore, the scheduling data utilized in this mission is presented in Table 13. Here, the parameters of maximum CPU time and maximum period are more restrictive, and execution windows were defined for payloads D and F. These parameters for execution window can be very useful for scheduling modules that require operation in very specific points along the orbit – for imaging or communication, for instance. Figure 5 shows the obtained schedule for each payload (organized from high to low priority) and respective power footprint. The computation time for the "Orbit 1" scenario was 390 milliseconds and for the "Orbit 2" scenario it was 62.2 seconds.

Table 13 – 2U nanosatellite payloads scheduling data.

Payload	j	u_j	q_j[w]	$y_j^{\min}$	$y_j^{\max}$	$t_j^{\min}$	$t_j^{\max}$	$p_j^{\min}$	$p_j^{\max}$	$w_j^{\min}$	$w_j^{\max}$
A	1	5	1	3	15	3	40	2	100	0	100
B	2	2	1.23	1	11	7	60	4	80	0	100
C	3	1	0.8	2	5	3	80	2	100	0	100
D	4	4	1.3	1	7	6	65	6	90	0	40
E	5	1	1.5	1	10	4	70	2	90	0	100
F	6	3	1.1	1	8	4	100	4	87	40	80
G	7	1	1.1	2	7	6	100	2	60	0	100
H	8	1	0.9	2	9	3	100	2	100	0	100

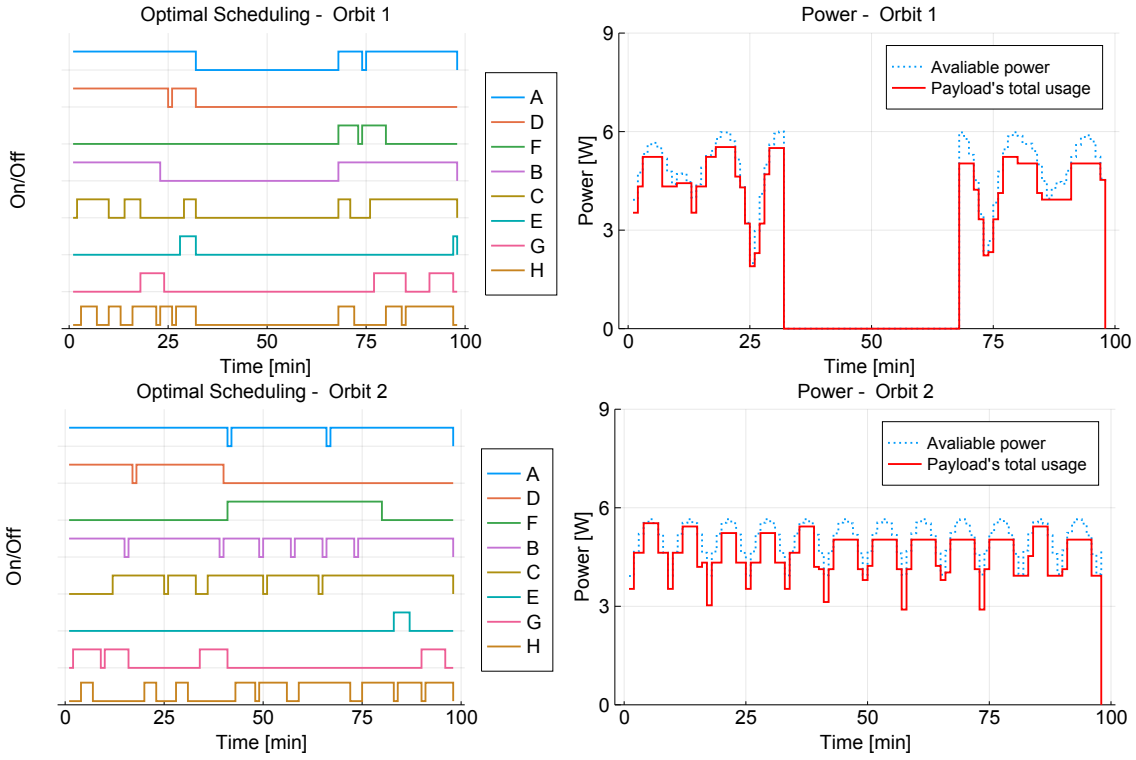
Source: **The author**.

From Figure 5 it can be seen that the 2U scheduling plan is considerably more sophisticated, with payloads of higher priority having their operation time maximized. Payload A, for instance, is only briefly switched off while payloads E and G are only momentarily turned on. It can also be observed that payloads D and F follow rigorously their execution windows. In addition, the resulting power usage follows more closely the available one, given the higher number of payloads and their wide-ranging energy impact. Here, in the "Orbit 1" scheduling, only 9.72% of the available power was not immediately utilized by the payloads, which could allow for a steady consumption of 0.80 W during the eclipse period.

3.3.2.3 3U Nanosatellite

Lastly, for the 3U nanosatellite, the previous example scheduling data was modified to consider payload A as an imaging module. This payload is estimated to occupy a 1U volume and to have a significant energy cost, as presented in Table 14. Parameters for execution window were also set for Payload A, given that imaging usually requires it – in this instance, payload

Figure 5 – 2U simulation results



Source: **The author**.

A could be expected to take 2 ($y_A^{\min}$), if possible 3 ($y_A^{\max}$), images when passing over a specific region of Earth. Figure 6 shows the resulted schedule for each payload (organized from high to low priority) and the respective power impact. Computation time for the "Orbit 1" scenario was 890 milliseconds and for the "Orbit 2" scenario it was 2.51 seconds.

Table 14 – 3U nanosatellite payloads scheduling data.

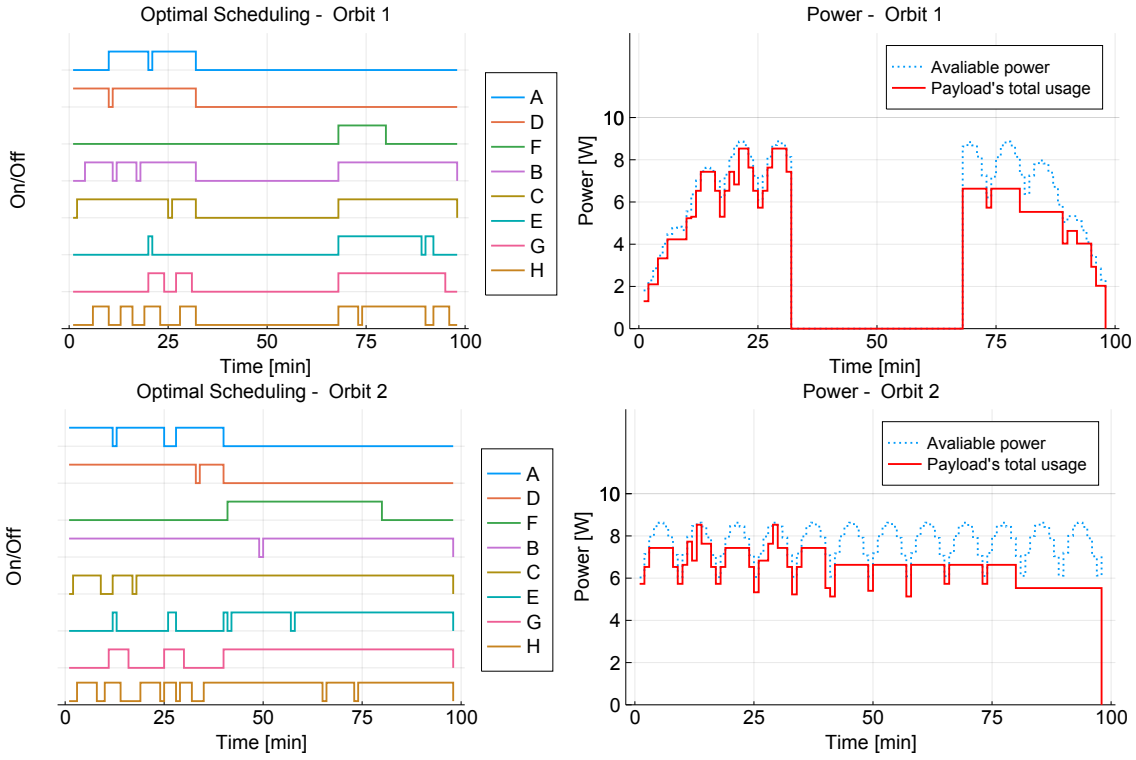
Payload	j	u_j	q_j[w]	$y_j^{\min}$	$y_j^{\max}$	$t_j^{\min}$	$t_j^{\max}$	$p_j^{\min}$	$p_j^{\max}$	$w_j^{\min}$	$w_j^{\max}$
A	1	5	3.2	2	3	10	12	5	78	0	40
B	2	2	1.23	1	4	2	50	4	50	0	100
C	3	1	0.8	2	3	3	80	2	100	0	100
D	4	4	1.3	1	7	6	65	6	90	0	40
E	5	1	1.5	1	10	1	70	2	60	0	100
F	6	3	1.1	1	8	2	100	4	87	40	80
G	7	1	1.1	2	3	4	100	2	60	0	100
H	8	1	0.9	3	9	3	100	2	40	0	100

Source: **The author**.

3.3.3 Computational impact

As previously mentioned, the guarantee of an optimal result, provided by MPM allied with the use of commercial solvers, has a notable computational impact. The preceding examples

Figure 6 – 3U simulation results

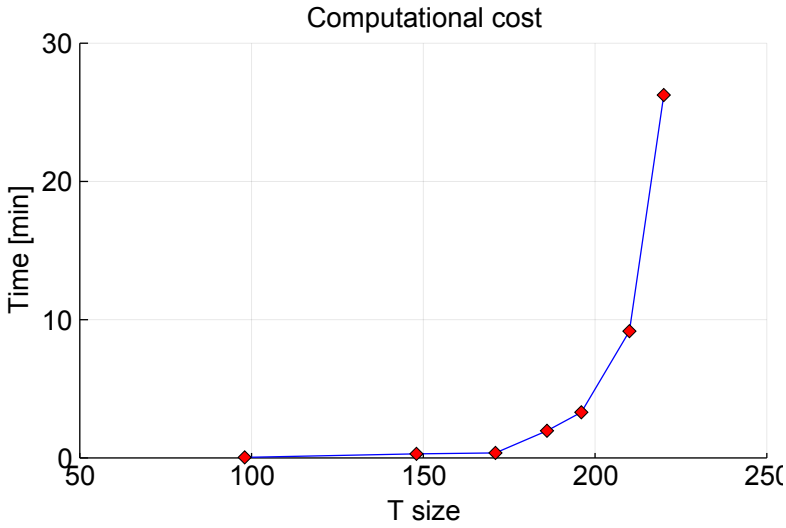


Source: **The author**.

have shown that the algorithm input data has an impact in solving time, be that in the form of task number, power vector, $\mathcal{T}$ size, or parameter restrictiveness. Of these factors, a larger $\mathcal{T}$ is of special interest because the scheduling problem could be solved for smaller time units (second by second, for example), allowing allocation of all tasks in the satellite, not just the payload activation moments.

In order to visualize this impact, considering the "Orbit 2" scenario of the 3U CubeSat, the simulation was repeated six more times but with $\mathcal{T}$ larger each time. To maintain the complexity of the scheduling problem and isolate the impact of this variable, the task parameters were adjusted proportionally to the increase in $\mathcal{T}$ and the energy vector was simply repeated. The results are presented in Figure 7, where an exponential increase in computational time can be seen as $\mathcal{T}$ gets larger. It becomes impractical to solve for larger instances, where the computational time for one orbit can take longer than the orbit period itself.

Figure 7 – Impact of $\mathcal{T}$ size on problem-solving time.



Source: **The author**.

3.4 DISCUSSION

In this chapter, it was presented a novel methodology for maximizing mission value and efficiency of nanosatellites by improving energy management with task scheduling optimization, constrained by solar irradiation levels reaching the photo-voltaic panels during an orbit. To that end, an existing analytical model was used for nanosatellite power input calculation according with two extreme scenarios of orbit inclination, two common attitudes for CubeSats, and three standard sizes.

Furthermore, a integer programming formulation was presented to maximize the number of tasks to be executed by a satellite, constrained by the power available minute by minute along the course of an orbit and task parameters such as priority, power footprint, minimum and maximum activation events, minimum and maximum execution times, minimum and maximum periods, and execution windows.

The scheduling results demonstrate that this combined approach generates an optimal energy effective mission plan while satisfying all task requirements, allowing the best possible harvesting and utilization of energy resources and, moreover, ensuring QoS.

However, the formulation lacks a battery model and does not easily optimize for more than one orbit period – high computational impact. The energy-neutral approach – where tasks with power consumption bigger than the power input are not allowed to execute – precludes tasks from executing in eclipse time. In addition, in a scenario where the tasks power consumption is smaller than the power input and the battery is not accounted for, harvested energy is wasted due to the model's inability to consider the unconsumed power for later use. The next chapter addresses these issues.

4 EXTENDED MULTI-ORBIT AND BATTERY FORMULATION

This chapter expands the core methodology of chapter 3 in two ways: to encompass a battery model and allow scheduling optimization for several orbits. The text of this chapter was published as an article in Rigo, Seman, Camponogara, Filho, et al. (2021b).

To model the battery, constraints for the State of Charge (SoC) were implemented following the *Coulomb Counting* method, in which the current flowing into and out of a battery is accounted for. When using this method, compensations for charging and discharging efficiency must be implemented to maintain accuracy (POP et al., 2005). Thus, a generalized disjunctive program was designed to select the efficiency constants. Fuzzy constraints were created to allow a degree of violation in energy resources utilization, representing the amount of energy that can be taken from the battery in a given moment. This, along with penalties for battery access ensures that the battery lifetime is preserved and even extended (LI et al., 2015; NING; HARAN; POPOV, 2003).

To present these contributions, this chapter is organized as follows. Section 4.1 consists of a review of fuzzy constraints and their applications. Sections 4.2 and 4.3 consists on the improvements to the optimization methodology previously presented. Section 4.4 presents the new power input calculation methodology. Notice that the power input methodology was not developed, nor was it implemented by the author, and is only presented in this thesis so the reader can fully understand the work. Section 4.5 reports on the results from the case studies. Section 4.6 draws some of the final remarks.

Firstly, the additional nomenclature used in this chapter is shown in Table 15.

4.1 FUZZY CONSTRAINTS

Since its inception in 1947 (BODINGTON; BAKER, 1990), Linear Programming (LP) has been widely applied in Operational Research primarily to maximize profits and reduce costs. However, one limitation of LP is the difficulty of modeling uncertainty, a characteristic inherent to real problems, particularly those that consider human behavior, environmental processes, transportation, economics, and fuzzy data in general (VILLACORTA et al., 2017).

The fundamental concepts of Fuzzy Linear Programming (FLP) were conceived in 1970 (BELLMAN; ZADEH, 1970) as a means to deal with the imprecision both in the coefficients and the constraints. Recently, FLP was applied to energy management (SADEGHI; HOSSEINI, 2013), transportation (SEMAN; RODRIGUES MACHADO, et al., 2020), supply chain planning under uncertainty (PISHVAEE; RAZMI, 2012; PEIDRO et al., 2009), logistics provider selection (WAN; WANG, et al., 2015), and software selection (WAN; QIN; DONG, 2017), to cite a few.

Fuzzy constraints are applied when a certain amount of violation in the restriction is permitted, as defined in (11),

$$a_k \mathbf{x} \leq_f b_k, \ \forall k \in \{1, \ldots, m\} \tag{11}$$

Table 15 – Added sets, constants and variables.

Notation	Definition
Sets	
$\mathcal{E}$	set of jobs that finished a previous orbit executing.
$\mathcal{W}$	set of jobs with no execution window limitations.
Constants	
V_b	battery nominal voltage, in Volts.
SoC_t	battery state of charge at time t, in %.
e_c	charge efficiency.
e_d	discharge efficiency.
Q	battery nominal capacity, in Ampere-hour.
ρ	minimum acceptable battery level, in %.
Variables	
b_t	decision variable that takes the value of 1 each time the charging rate should be used, and 0 each time the discharging rate should be used.
α_t	decision variable that takes values between 0 and 1 representing the amount of energy drawn from the battery.
k_t	difference between power input and power consumption at time t, in Watts.
i_t	battery current, in Ampere.
a_t	assumes the values of $\lvert i_t \rvert$.

Source: **The author**.

and modeled in the membership function (12),

$$\mu_k : \mathbb{R} \to [0,1], \;\; \mu_k(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \leq b_k \\ f_k(\mathbf{x}), & \text{if } b_k \leq \mathbf{x} \leq b_k + h_k \\ 0, & \text{if } \mathbf{x} \geq b_k + h_k \end{cases} \tag{12}$$

where $\mathbf{x}$ is the decision variable, a_k is a matrix and b_k is a vector, both obtained from the m constraints. In other words, f_{b_k} can assume values ranging from b_k to $b_k + h_k$, these being the bounds of the fuzzy distribution.

From the membership function (12), a degree of violation of at most $b_k + h_k$ is permitted for $\mathbf{x}$ in the k-th constraint, and the problem to be solved becomes the one given in (13).

$$\max z = c\mathbf{x} \tag{13a}$$

$$\text{s.t. } a_k\mathbf{x} \leq_f b_k, \; \forall k \in \{1, \ldots, m\} \tag{13b}$$

$$\mathbf{x} \geq 0, \;\; \mathbf{x} \in \mathbb{R} \tag{13c}$$

To resolve the fuzzy constraint (13b), four of the most popular approaches from literature (SAFI; MALEKI; ZAEIMAZAD, 2007) are introduced below. However, for all cases $f_k(\mathbf{x})$ must be linear.

4.1.1 Zimmermann's method

Zimmermann (1978) introduces a new objective (λ) and adds the former ($c\mathbf{x}$) to the

constraints, making it a fuzzy goal, as formalized in (14).

$$\max \lambda \tag{14a}$$

$$\text{s.t. } c\mathbf{x} \geq b_0 - h_0\ (1-\lambda) \tag{14b}$$

$$A_k\mathbf{x} \leq b_k + h_k\ (1-\lambda),\ \forall k \in \{1,\ldots,m\} \tag{14c}$$

$$\mathbf{x} \geq 0,\ \ \lambda \in [0,1] \tag{14d}$$

Here, b_0 is an aspiration level chosen by the decision-maker; consequently, Zimmermann's method does not guarantee optimality and can be unbounded (SAFI; MALEKI; ZAEIMAZAD, 2007). Nonetheless, his approach can be utilized to minimize the degree of violation of the constraints.

4.1.2 Werners's method

Werners (1987) proposes a method to find the extreme solutions of the problem, so that the decision-maker does not need to specify a goal or a tolerance. It consists of a particular case of the Zimmermann's method, where:

$$b_0 = \sup\{\max_{\mathbf{x}\in\mathcal{X}} c\mathbf{x}\} \tag{15a}$$

$$h_0 = \sup\{\max_{\mathbf{x}\in\mathcal{X}} c\mathbf{x}\} - \inf\{\max_{\mathbf{x}\in\mathcal{X}} c\mathbf{x}\} \tag{15b}$$

where $\mathcal{X} = \{\mathbf{x} \in \mathbb{R} : a_k\mathbf{x} \leq_f b_k, \mathbf{x} \geq 0\}$.

4.1.3 Tanaka's method

Tanaka, Okuda, and Asai (1973) normalizes the objective function ($c\mathbf{x}$) by the optimum value (Z) when solving the problem with crisp constraints, as presented in (16).

$$\max \lambda \tag{16a}$$

$$\text{s.t. } \frac{c\mathbf{x}}{Z} \geq \lambda \tag{16b}$$

$$A_k\mathbf{x} \leq b_k + h_k\ (1-\lambda),\ \forall k \in \{1,\ldots,m\} \tag{16c}$$

$$\mathbf{x} \geq 0,\ \ \lambda \in [0,1] \tag{16d}$$

Here, Z serves as an aspirational level. Notice that Tanaka's method can be reduced to Zimmerman's method by setting $b_0 = h_0 = Z$ in (14) (VILLACORTA et al., 2017).

4.1.4 Verdegay's method

Verdegay (1982) generalizes the Zimmermann's and Tanaka's method, proving that their solutions for an FLP are particular values of the Verdegay's fuzzy solution. His method consists of a parametric approach to solve an FLP problem with fuzzy constraints, which is formalized here in (17).

$$\max z = c\mathbf{x} \tag{17a}$$

$$A_k\mathbf{x} \leq b_k + h_k\ (1-\lambda),\ \forall k \in \{1,\ldots,m\} \tag{17b}$$

$$\mathbf{x} \geq 0,\ \ \lambda \in [0,1] \tag{17c}$$

By solving the LP problem (17), an optimal solution can be found for each λ-cut, resulting in a fuzzy solution for the original fuzzy problem (13).

4.2 PROPOSED BATTERY MODEL

This chapter proposes to introduce a battery model into the scheduling framework. First, consider that the state-of-charge of a generic battery is given by:

$$\mathrm{SoC}(t) = \frac{1}{Q}\int i(t)dt \tag{18}$$

For this model, the state-of-charge of the battery can be represented as a Generalized Disjunctive Program (GDP) (GROSSMANN; LEE, 2003), as shown in (19).

$$\begin{bmatrix} b_t \\ \mathrm{SoC}_{t+1} = \mathrm{SoC}_t + \frac{i_t \cdot e_c}{60\ Q} \end{bmatrix} \bigvee \begin{bmatrix} \neg\ b_t \\ \mathrm{SoC}_{t+1} = \mathrm{SoC}_t + \frac{i_t \cdot e_d}{60\ Q} \end{bmatrix}, \forall t \in \mathcal{T}, \tag{19a}$$

$$b_t \in \Big\{True, False\Big\}, \forall t \in \mathcal{T}, \tag{19b}$$

If b_t is true, then the constraint with the charging efficiency is selected (e_c), otherwise the constraint with the discharging efficiency is activated (e_d). The battery current i_t is estimated by (20a), using the power calculated by (20b). Notice that the power k_t is either positive or negative, depending on whether the satellite is consuming or supplying power to the battery.

$$i_t = \frac{k_t}{V_b}, \qquad \forall t \in \mathcal{T}, \tag{20a}$$

$$k_t = r_t\ - \sum_{s=1}^{S}\sum_{j=1}^{J} q_{sj} x_{sjt}, \qquad \forall t \in \mathcal{T}, \tag{20b}$$

Then, the GDP presented in (19) is transformed into a MILP formulation, following the commonly used Big-M methodology (CASTRO, 2015) expressed in constraints (21).

$$\mathrm{SoC}_{t+1} \geq \mathrm{SoC}_t + \frac{i_t \cdot e_c}{60\ Q} - M \cdot (1-b_t), \qquad \forall t \in \mathcal{T}, \tag{21a}$$

$$\mathrm{SoC}_{t+1} \leq \mathrm{SoC}_t + \frac{i_t \cdot e_c}{60\ Q} + M \cdot (1-b_t), \qquad \forall t \in \mathcal{T}, \tag{21b}$$

$$\mathrm{SoC}_{t+1} \geq \mathrm{SoC}_t + \frac{i_t \cdot e_d}{60\ Q} - M \cdot b_t, \qquad \forall t \in \mathcal{T}, \tag{21c}$$

$$\mathrm{SoC}_{t+1} \leq \mathrm{SoC}_t + \frac{i_t \cdot e_d}{60\ Q} + M \cdot b_t, \qquad \forall t \in \mathcal{T}, \tag{21d}$$

$$i_t \leq M \cdot b_t, \qquad \forall t \in \mathcal{T}, \tag{21e}$$

$$i_t \geq -M \cdot (1-b_t), \qquad \forall t \in \mathcal{T}, \tag{21f}$$

$$b_t \in \{0,1\}, \qquad \forall t \in \mathcal{T}, \tag{21g}$$

Considering these enhancements to the battery model, its assumed that the available power follows a fuzzy behavior, as shown in:

$$\sum_{s=1}^{S}\sum_{j=1}^{J} q_{sj}x_{sjt} \leq_f r_t \tag{22}$$

which can be modeled as:

$$\mu_k : \mathbb{R} \to [0,1], \;\; \mu_k(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \leq r_t \\ f_k(\mathbf{x}), & \text{if } r_t \leq \mathbf{x} \leq r_t + \gamma V_b \\ 0, & \text{if } \mathbf{x} \geq r_t + \gamma V_b \end{cases} \tag{23}$$

where γ is the maximum current allowed for the battery charge or discharge (C and E rates). Limiting these rates has been shown to preserve the battery lifetime (NING; HARAN; POPOV, 2003).

In order to include the fuzzy model into the optimization formulation, Verdegay's method is followed, as detailed in (24a) and (24b), allowing the task schedule to exceed the maximum energy limit previously imposed by the solar panels. Now, power can be also supplied from the battery. In other words, the input power can range from r_t (total power input by solar panels at time t) to $r_t + \gamma V_b$. A decision variable $\alpha_t \in [0,1]$ is used to decide the energy drawn from the battery; when $\alpha_t = 1$ no power is supplied by the battery, and when $\alpha_t = 0$, a total of γV_b units of power are used.

$$\sum_{s=1}^{S}\sum_{j=1}^{J} q_{sj}x_{sjt} \leq r_t + \gamma\ V_b\ (1-\alpha_t), \qquad \forall t \in \mathcal{T}, \tag{24a}$$

$$0 \leq \alpha_t \leq 1, \qquad \forall t \in \mathcal{T}, \tag{24b}$$

In an attempt to maximize the cost function related to task scheduling, the optimization model — solved by an off-the-shelf solver (i.e., Gurobi) — will have degrees of freedom to find the optimal value of α while respecting the remaining restrictions of the problem.

Additionally, constraints are imposed in order to guarantee that the SoC remains within acceptable limits:

$$\mathrm{SoC}_T \leq \mathrm{SoC}_1(1+\Delta^{\mathrm{SoC}}) \tag{25a}$$

$$\mathrm{SoC}_T \geq \mathrm{SoC}_1(1-\Delta^{\mathrm{SoC}}) \tag{25b}$$

$$\mathrm{SoC}_t \leq 1,\ \forall t \in \mathcal{T}, \tag{25c}$$

$$\mathrm{SoC}_t \geq \rho,\ \forall t \in \mathcal{T}, \tag{25d}$$

Constraints (25c) ensure that energy is not wasted by saturating the battery, while constraints (25a) and (25b) ensure that all input energy is utilized within the orbit, by stating that

the state-of-charge at the last orbit moment (SoC_T) must be no less and no more than 1% of the state-of-charge at the first orbit moment (SoC_1). The relevance of this constraint will be further explained in the results section. In constraints (25d), ρ is the minimum acceptable battery level that limits the depth-of-discharge, which can also extend battery lifetime (LI et al., 2015).

Finally, a new objective function (26) is designed in order to penalize battery usage.

$$F: \max_{x_{sjt}} \sum_{s=1}^{S}\sum_{j=1}^{J}\sum_{t=1}^{T} u_{sj}x_{sjt} - \beta\sum_{t=1}^{T} a_t \tag{26}$$

where β is the penalty weight, and constraints (27a) through (27c) ensure that the variable $a_t \in \mathbb{R}$ will assume the value $|i_t|$.

$$a_t \geq i_t, \qquad \forall t \in \mathcal{T}, \tag{27a}$$

$$a_t \geq -i_t, \qquad \forall t \in \mathcal{T}, \tag{27b}$$

$$a_t \in \mathbb{R}, \qquad \forall t \in \mathcal{T}, \tag{27c}$$

4.3 ORBIT COUPLING

The formulation of Chapter 3 allows tasks with no execution window limitations to start at the end of an orbit. This means that, at the beginning of the next orbit, the job must finish its minimum execution and period time, a condition that is not addressed. In this section, modifications are developed to the original formulation to allow the continuity and consistency of the task schedule for more than one orbit period. This is achieved without the need of new variables, which could increase the solution time, but rather by introducing three coupling constraints and further conditioning and/or modifying the scope of others.

Let $\mathcal{W}$ be a set of jobs with no execution window limitations. Two vectors bring historical data about the past orbit: τ_{sj} indicates if the job j finished the last orbit in the running mode, in which case τ_{sj} assumes value 1, otherwise τ_{sj} assumes value 0; and ε_{sj} indicates how many units of time elapsed before time T since the last job activation, $\phi_{sj} = 1$. Furthermore, let $\mathcal{E}$ be a set comprised of jobs that finished the last orbit executing. Constraint (28) was formulated to be imposed when a job terminated the previous orbit executing, which states that the sum of x at the beginning of the next orbit must assume a value between the minimum and maximum executing time of that job, minus what was already executed in the previous orbit (namely $-\varepsilon_{sj}$).

$$t_{sj}^{\min} - \varepsilon_{sj} \leq \sum_{t=1}^{t_{sj}^{\max}-\varepsilon_{sj}+1} x_{sjt} \leq t_{sj}^{\max} - \varepsilon_{sj}, \ \forall j \in \mathcal{E}, \ \forall s \in \mathcal{S}, \tag{28}$$

Constraint (29) was created to guarantee the initial minimum period for all $j \in \mathcal{W}$, stating that all ϕ at the beginning of the next orbit must assume a value zero until the minimum period of the job, minus what was already executed (namely $-\varepsilon_{sj}$).

$$\phi_{sjt} = 0, \ \forall t = \tau_{sj}+1, \ldots, p_{sj}^{\min} - \varepsilon_{sj}, \forall j \in \mathcal{W}, \forall s \in \mathcal{S}, \tag{29}$$

Constraint (30) was formulated to ensure that, after the initial period is fulfilled, the job starts executing again.

$$\sum_{\tau_{sj}+1}^{p_{sj}^{\max}-\varepsilon_{sj}} \phi_{sjt} \geq 1, \forall j \in \mathcal{W}, \forall s \in \mathcal{S}, \tag{30}$$

The scope of (6a) is modified so that it is no longer valid from $t = 1$, but from $t = \tau_{sj} + 1$, as in (31). This is needed because x_{sj1} is forced to 1, and $\phi_{sj1} = 1$, triggering (6a) to enforce a full run time t_{sj}, when the job should only execute for $t_{sj} - \varepsilon_{sj}$.

$$\sum_{l=t}^{t+t_{sj}^{\min}-1} x_{sjl} \geq t_{sj}^{\min} \phi_{sjt}, \forall t \in \left\{\tau_{sj}+1, \ldots, T - t_{sj}^{\min}+1\right\}, \forall j \in \mathcal{W}, \forall s \in \mathcal{S}, \tag{31}$$

Similarly, the scope of (7a) and (7b) are modified to (32a) and (32b), respectively, so that they are valid from $t = p_{sj}^{\min} - \varepsilon_{sj}$ avoiding conflict with (29) and (30).

$$\sum_{l=t}^{t+p_{sj}^{\min}-1} \phi_{sjl} \leq 1, \forall j \in \mathcal{W}, \forall s \in \mathcal{S}, \forall t \in \left\{p_{sj}^{\min} - \varepsilon_{sj}+1, \ldots, T - p_{sj}^{\min}+1\right\} \tag{32a}$$

$$\sum_{l=t}^{t+p_{sj}^{\max}-1} \phi_{sjl} \geq 1, \forall j \in \mathcal{W}, \forall s \in \mathcal{S}, \forall t \in \left\{p_{sj}^{\max} - \varepsilon_{sj}+1, \ldots, T - p_{sj}^{\max}+1\right\} \tag{32b}$$

In constraints (32a) and (32b) the value of $p_{sj} - \varepsilon_{sj}$ can be greater than $T - p_{sj} + 1$, for a large p_{sj} or a small T, which would let ϕ free for the rest of the orbit. Constraints (33a) and (33b) were created to address this condition, ensuring the minimum and maximum period at the end of the orbit.

$$\sum_{l=t}^{T} \phi_{sjl} \leq 1, \qquad \forall t \in \left\{T - p_{sj}^{\min}+1, \ldots, T\right\}, \forall j \in \mathcal{W}, \forall s \in \mathcal{S}, \tag{33a}$$

$$\sum_{T-p_{sj}^{\max}+1}^{T} \phi_{sjt} \geq 1, \qquad \forall j \in \mathcal{W}, \forall s \in \mathcal{S}, \tag{33b}$$

Finally, Algorithm 1 is invoked for all orbits greater than 1 to condition the application of the new constrains, along with the modifications to the constraints of the baseline formulation. Constraints (4), (5), (6b), (6c), (8), and (9) are applied without modifications.

Note the logic $x_{sj1} \geq \operatorname{sgn}(t_{sj}^{\min} - \varepsilon_{sj})\, \tau_{sj}$. If $\tau_{sj} = 0$, then job j is free to execute at the beginning of the new orbit or not ($x_{sj1} \geq 0$); the same occurs if $\tau_{sj} = 1$ and the job has already finished executing its minimum required time in the previous orbit, which leads to $-x_{sj1} \geq -1 \implies x_{sj1} \leq 1$. However, if $\tau_{sj} = 1$ and the job has not finished executing its minimum required time in the previous orbit, then $x_{sj1} \geq 1$, forcing the job to continue running, being subject to (28).

Algorithm 1: Orbit Coupling

```
for s ∈ 𝒮, j ∈ 𝒥 do
    if j ∈ 𝒲 then
        Impose x_sj1 ≥ sgn(t_sj^min − ε_sj) τ_sj
        if τ_sj = 1 then
            Impose (28)
        Impose (29), (30), (31), (32), and (33)
    else
        Impose (6a) and (7)
```

4.3.1 Solving the optimization problem

The optimal schedule of a nanosatellite tasks is achieved by solving the problem formally defined below:

$$F: \max_{x_{sjt}} \sum_{s=1}^{S}\sum_{j=1}^{J}\sum_{t=1}^{T} u_{sj}x_{sjt} - \beta\sum_{t=1}^{T} a_t$$
$$\text{s.t. } (4)\text{–}(9), (20), (21), (24), (25), (27)\text{–}(33)$$

Notice that the resulting problem is a MILP problem, which has the general form of (34).

$$\min c^T\mathbf{x} + h^T\mathbf{y} \tag{34a}$$
$$\text{s.t. } A_k\mathbf{x} + G_k\mathbf{y} \leq b_k,\ \forall k \in \{1,\ldots,m\} \tag{34b}$$
$$\mathbf{x} \in \mathbb{Z}^n, \mathbf{y} \in \mathbb{R}^p \tag{34c}$$

where the coefficients of the vectors c and h are easily obtained from (26), A and G are matrices and b is a vector which are all obtained from the m constraints aforementioned.

MILP constitutes a general class of NP-Hard problems, in the sense that any NP-Complete problem (e.g., the classic satisfiability problem) can be reduced to MILP in polynomial time. Any polynomial time algorithm for MILP would imply that P=NP (GAREY; JOHNSON, 1990); in other words, all decision problems would be solved in polynomial time. MILP problems can possibly be solved to optimality by exact algorithms, such as the branch-and-bound (VANDERBEI, 2001) and cutting-plane methods (MARCHAND et al., 2002), and approximately using heuristics. Many commercial and non-commercial solvers, like Gurobi (GUROBI OPTIMIZATION, 2016) and CBC (FORREST et al., 2018), respectively, can be employed to obtain optimal solutions or solutions with a certificate of quality.

4.4 POWER INPUT

For simulation purposes, the varying power input, calculated with Equation 1 (FILHO et al., 2020), was based on two-line element data of the CubeSat FloripaSat-I, operating in

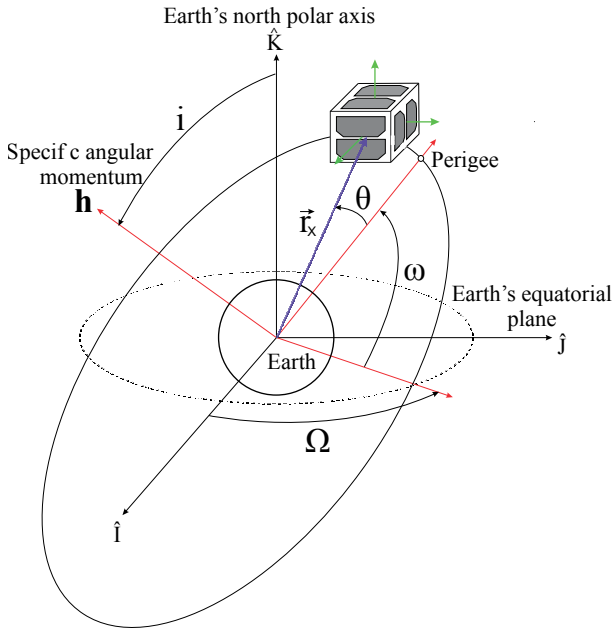
an orbit with J2 perturbation and an attitude that keeps one face of the nanosatellite towards the Earth for the entire orbit, similar to a remote sensing mission. Notice that the power input methodology was not developed, nor was it implemented by the author, and is only presented in this thesis so the reader can fully understand the work.

4.4.1 Orbit model

To make it useful for the design and analysis of typical CubeSat missions, the algorithm for orbit determination is structured to have Two Line Element (TLE) data as input, namely: Inclination [°]; Ω: right ascension of the ascending node [°]; e: eccentricity [-]; ω: argument of perigee [°]; M: mean anomaly [°]; n: mean motion [rev/day].

Figure 8 shows a schematic view of the orbit and some of these parameters.

Figure 8 – Orbit diagram.



Source: Modified from Filho et al. (2020).

The values in the TLE give rise to other variables that enable the calculation of the CubeSat's position in the perifocal frame of reference, projection the orbit plane, and then provides the satellite's position in the geocentric equatorial frame of reference. For further details, see Curtis (2014). Because of the J_2 perturbation, caused by the oblateness of the Earth, the angles Ω and ω change with time, at very small rates. As a consequence, the time spent under the Earth's shadow may vary.

4.4.1.1 Attitude Model

The attitude model aims to mimic the rotation of the satellite around its own axis. For a typical CubeSat geometry, without regard of its size (1U, 2U, 3U, ...), without deployable solar

panels, there are six normal vectors $\vec{n}_w$ that describe the orientation of each side w, as evidenced in Figure 8 by the green arrows.

To add the spin on the satellite, rotation matrices are built as a function of the angular speed and axis of rotation to project the normal vectors onto new orientations.

4.4.2 Studied scenarios

A typical CubeSat is comprised of three main modules – an On-Board Data Handler (OBDH), a Telemetry, Tracking and Command (TT&C) module and the Electrical Power System (EPS) – and the payloads, which effectively give them functionality or value. The nanosatellite size will be the limiting factor for how many payloads can be accommodated in a given mission; a 1U CubeSat usually accommodates six printed circuit boards (MARCELINO et al., 2020). Bigger size also means more energy input, since there is more surface area for energy harvesting.

The power input represents the energy available at the solar panel outputs, so the data is multiplied by an EPS efficiency of 0.85. This efficiency is that of the FloripaSat-I mission, a 1U CubeSat developed by students at the Federal University of Santa Catarina (UFSC) and launched in 2019 (MARCELINO et al., 2020). The TLE of FloripaSat-I is used as input for the orbit model, as listed in Table 16.

Table 16 – TLE input for the case studies.

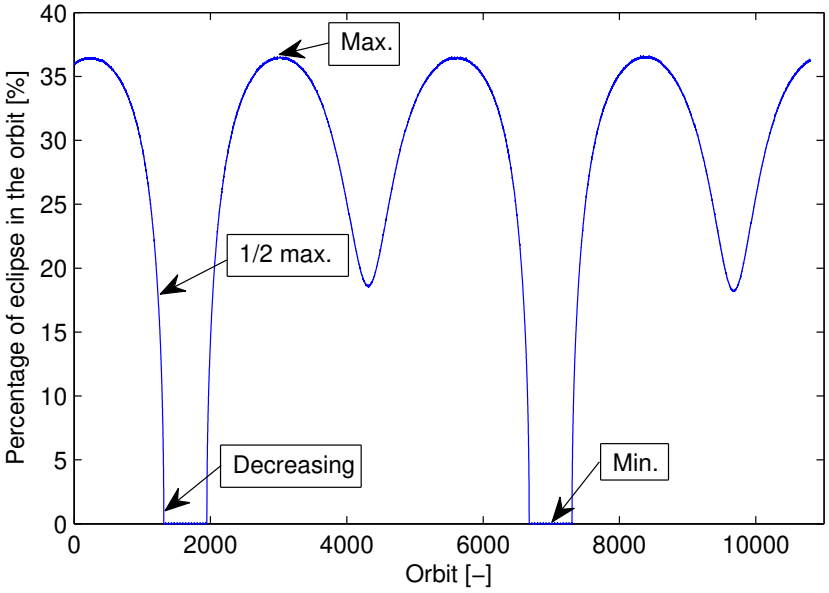
CubeSat	i	Ω	e	ω	M	n
FloripaSat-I	97.95	225.78	0.0016	111.38	248.91	14.82

Source: **The author**.

The satellite keeps one face towards the center of the Earth along the entire orbit, simulating a remote sensing mission. As a result from the orbit model with perturbation J_2, the orbit plane slightly changes with time and creates different scenarios with access to the solar radiation. The ratio of time spent under the shadow of the Earth and the period of the orbit is shown in Figure 9, for each orbit of FloripaSat-I along a year. The orbits with maximum eclipse have around 36% of absence of irradiance, or in other words, around 35 minutes without external power input. On the other hand, there are weeks where the satellite will not face any shadow of the Earth, so it could operate at maximum capacity.

In this work, the analysis will be performed for consecutive orbits in the regions indicated by *Max.* and 1/2 *Max.* in Figure 9. The condition with minimum eclipse does not introduce transient constraints in the power input of the satellite, so the optimization is not appreciable in that region.

Figure 9 – Fraction of eclipse for FloripaSat-I orbit along a year.



Source: Rigo, Seman, Camponogara, Filho, et al. (2021b).

4.5 RESULTS AND ANALYSIS

The methodologies presented in Subsections 4.2 and 4.3 were implemented in the Julia programming language using the JuMP library (DUNNING; HUCHETTE; LUBIN, 2017) and solved using the Gurobi solver (GUROBI OPTIMIZATION, 2016) in a PC with an Intel(R) Core(TM) i7-8550U 1.8 GHz processor, 16 GB of RAM, and Windows 10 64 bits.

In order to show the effectiveness of the methodology, three scenarios are explored here.

- **Scenario A** considers a 3U mission in an orbit with a half-maximum eclipse and three cases: one where no battery is modeled – without fuzzy constraints, for comparison purposes – and two cases with energy management using batteries.
- **Scenario B** refers to the same 3U mission, except for scheduling the satellite tasks for an orbit of maximum eclipse.
- **Scenario C** addresses the requisites of a 6U mission.

4.5.1 Requisites: 3U mission

The scheduling framework previously established is applied here to determine at which moments the payloads of a 3U nanosatellite can be activated, in a minute-by-minute basis, for several orbits to maximize the mission value while preserving battery life and fulfilling mission requirements. Table 17 gives the scheduling data used as input, representing a 3U CubeSat mission, and Table 18 presents the parameters values for the experiments, unless otherwise stated.

Table 17 – Nanosatellite jobs data.

	A	B	C	D	E	F	G	H	I
j	1	2	3	4	5	6	7	8	9
u_{sj}	5	2	1	4	1	3	1	2	6
q_{sj}	3.2	1.23	1.8	1.3	2.5	1.4	2.1	0.7	0.4
$y_j^{\min}$	2	2	1	1	1	1	1	1	1
$y_j^{\max}$	3	4	6	7	5	6	7	8	1
$t_j^{\min}$	5	16	17	14	18	13	16	17	T
$t_j^{\max}$	6	25	20	32	35	18	30	22	T
$p_j^{\min}$	8	28	25	35	38	25	32	25	T
$p_j^{\max}$	32	52	49	59	62	49	56	49	T
$w_j^{\min}$	30	0	0	0	0	40	0	0	0
$w_j^{\max}$	80	T	T	T	60	80	T	T	T

Source: **The author**.

Note that jobs from 1 to 8 (or A to H) represent the mission payloads, and job 9 (or I) represents all core modules of the satellite. Here the jobs are assumed to be active at all time and consume 400 mW (MARCELINO et al., 2020). Furthermore, for illustration purposes, payload A corresponds to an image sensing module, which must take a minimum number of images when passing over a specific region of the Earth, here specified by the execution window and minimum activation parameters.

Table 18 – Scheduling parameters.

S	J	T	Q	V_b	e_c	e_d	SoC_0	ρ	γ	β
1	9	97	5 Ah	3.6 V	0.95	0.8	75%	0.5	5 A	10

Source: **The author**.

In Table 18, T is the period of the chosen orbit (in minutes), whereas Q is a typical battery capacity for a 3U mission. The values of V_b, e_c, and e_d were obtained from the data-sheet of the battery used in the FloripaSat-I mission (SAMSUNG SDI CO., 2014). To guarantee that the battery would never discharge more than 50%, a common requirement in critical applications, ρ was set to 0.5. Finally, γ was set to 5 A, or 1 C, since greater discharge rates can compromise the battery lifetime (NING; HARAN; POPOV, 2003).

4.5.1.1 Scenario A – 1/2 Max. Orbit

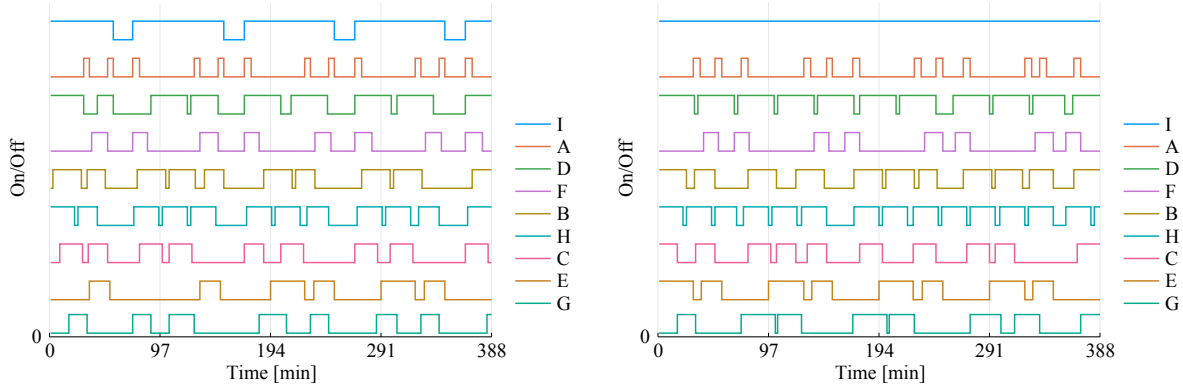
Scenario A considers the 3U mission requisites with an orbit of half-maximum eclipse. For this scenario three cases are considered:

- A1: no battery considered (without fuzzy constraints) (RIGO; SEMAN; CAMPONOGARA; MORSCH FILHO; BEZERRA, 2021);

- A2: fuzzy constraints with battery access limited by constraints (25a) and (25b), , with $\Delta^{\text{SoC}} = 1\%$;
- A3: fuzzy constraints without battery access limits.

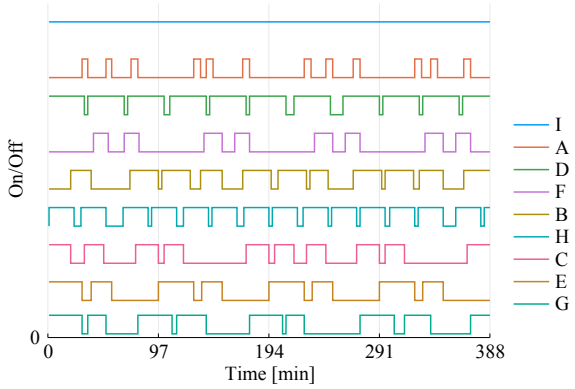
For the first case (A1) with data from Table 17, where no battery is modeled, the optimization is infeasible given that the core modules cannot execute during eclipse time. Therefore, $t_9^{\min}$ was set to 20 minutes and $y_9^{\min}$ to 2 minutes, in order to obtain a feasible schedule. The resulting schedule for four consecutive orbits is shown in Figure 10a, where the job activation and deactivation times are plotted in order of priority. For all cases, notice that the x-axis grid indicates when one orbit ends and the next begins (every 97 minutes).

Figure 10 – Scheduling results for a 3U satellite in a half-maximum eclipse orbit.



(a) Optimal scheduling for scenario A case 1 – A1 (no battery).

(b) Optimal scheduling for scenario A case 2 – A2.



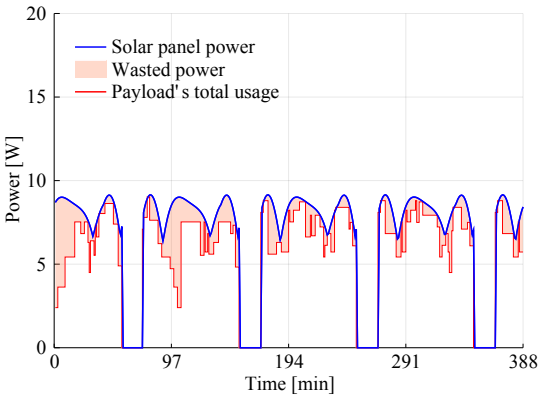
(c) Optimal scheduling for scenario A case 3 – A3.

Source: **The author**.

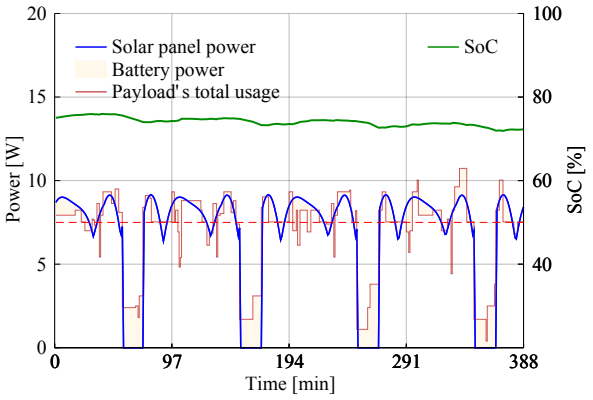
It can be seen that job I execution was maximized to run continuously, given its high priority, however the satellite would be completely off during the eclipse time, given the lack of energy. Notice that payload A executes only within its execution window, whereas payload B continues to run along the orbit, complying with its execution requirements and showing the effectiveness of the orbit coupling methodology. The same cross orbit behavior occurs with payloads D and G.

The power analysis for this schedule is shown in Figure 11a. Note that the solar panel power is calculated with the methodology presented in Section 4.4. Despite the optimal solution of the scheduling problem, the total power consumption cannot perfectly match the input power, therefore wasting energy.

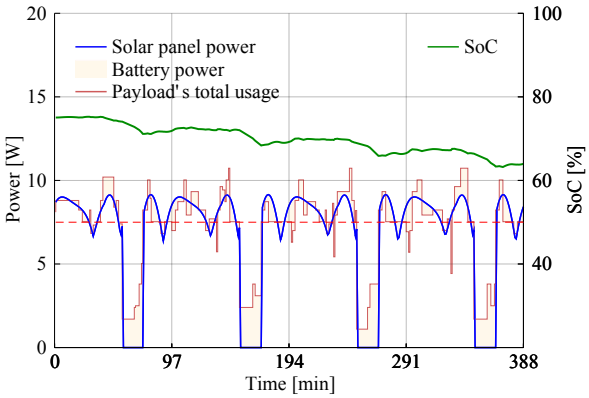
Figure 11 – Energy analysis for a 3U satellite in a half-maximum eclipse orbit.



(a) Energy analysis for scenario A case 1 – A1.



(b) Energy analysis for scenario A case 2 – A2.



(c) Energy analysis for scenario A case 3 – A3.

Source: **The author**.

For case 2 (A2), the full scheduling methodology aforementioned is applied, and the resulting schedule for four consecutive orbits is presented in Figure 10b. Now, job I runs continuously without interruption, ensuring that the satellite core modules will be active at all times. It can be observed that all task requirements are fulfilled, even across orbits. Payloads E and F, for instance, enter a fully periodic operation regime. Figure 11b shows the power analysis for this case. Note that the battery power indicates both the energy being consumed or provided to the battery. Here the power consumption follows the power input closely, a behavior expected given the battery access penalty and constraints (25a) and (25b). Nevertheless, the battery is used mainly to power the core modules during the eclipse time, or to run the tasks more effectively, such as to follow execution window parameters.

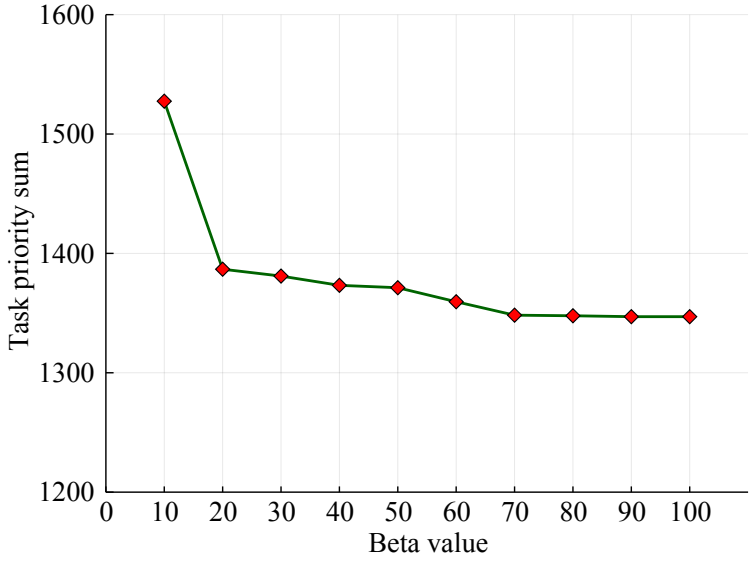
Although cyclic regime emerges in this example, the multi-orbit formulation also serves to deal with tasks that started at the end of one orbit and must end at the beginning of another. Even

in a periodical regime, these boundary conditions from the previous orbits must be accounted for in the optimization of the scheduling problem. Notice that there may be a change in the battery's energy level, forcing the periodic payload regime to change over time to render the optimization problem feasible. In addition to being closely related to the execution of tasks, the battery level is dependent on the input energy available in the satellite. In closed orbits, the variation in the input is not perceptible. However, when analyzing the satellite's useful life, as shown by Filho et al. (2020), there may be a significant variation in the energy input levels due to disturbances in orbit.

In case 3 (A3), the scheduling formulation is free to access the battery energy, by dropping the constraints (25a) and (25b). The optimal schedule is presented in Figure 10c, however the impact of these changes is significant in the energy footprint, as presented in Figure 10c. The battery usage often peaks at more than 10 W and, as a consequence, the battery SoC undergoes more variation when compared to case A2, rapidly falling towards the ρ limit.

Finally, Figure 12 shows the variation in the portion of the cost function relative to the execution of tasks – cost function (2) – in relation to the β penalty. The plot reveals that the execution of tasks decreases with the increase in the penalty for battery usage, until finally stabilizing for high values of β.

Figure 12 – β variation impact on task priority sum – cost function (2).



Source: **The author**.

4.5.1.2 Scenario B – Max. Orbit

Scenario B refers the 3U mission requisites with a maximum eclipse orbit. This scenario consists of three cases:

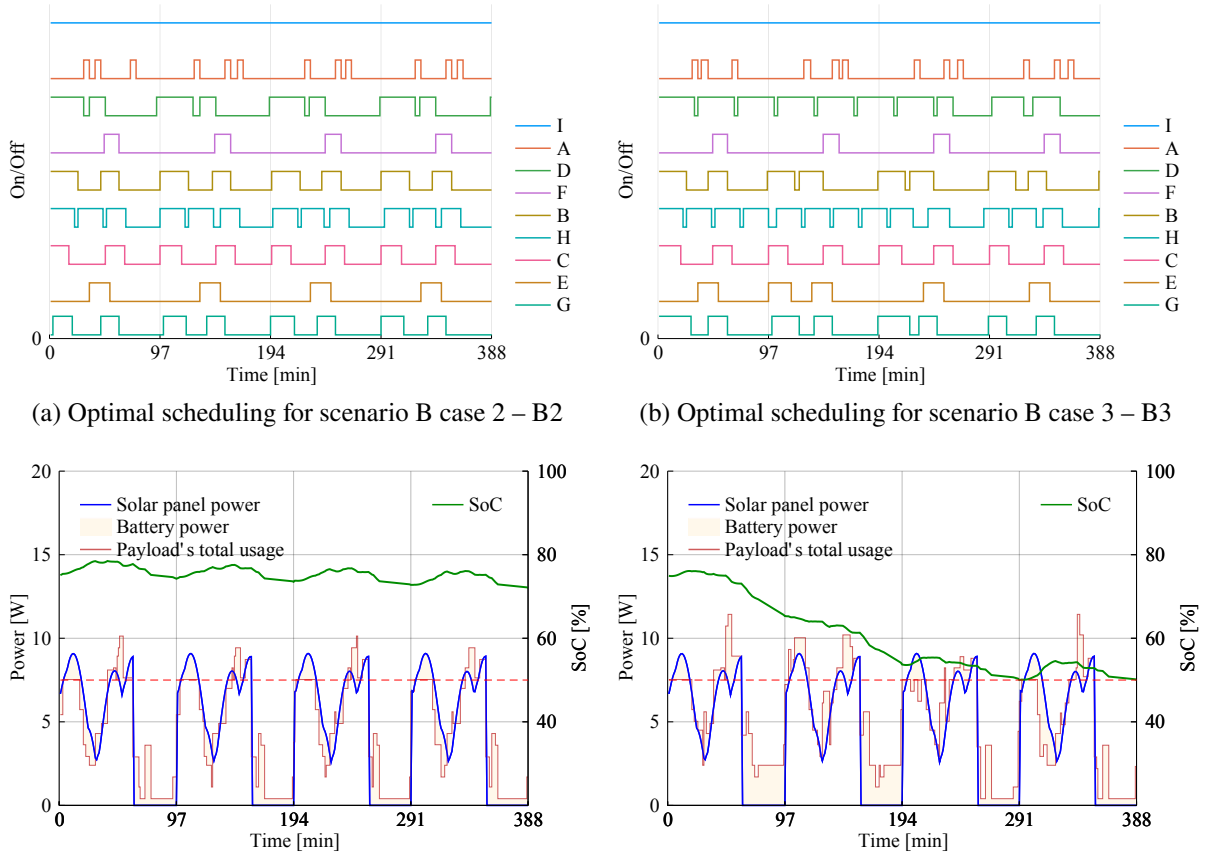
- B1: no battery considered (without fuzzy constraints) (RIGO; SEMAN; CAMPONOGARA; MORSCH FILHO; BEZERRA, 2021);

- B2: fuzzy constraints with battery access limited by constraints (25a) and (25b), with $\Delta^{\text{SoC}} = 1\%$;
- B3: fuzzy constraints with no battery access limits.

This scenario explores an orbit variation, with a significant increase in eclipse time if compared to the scenario A. For this orbit, the same cases of the previous scenario are analyzed; however, despite the modifications in the data setup to enable case 1 of scenario A, the optimization problem was not feasible for this orbit where energy is scarce. In other words, case B1 is infeasible without accounting for a battery.

For case 2 (B2), considering the battery, the obtained schedule is illustrated in Figure 13a.

Figure 13 – Scheduling results for a 3U satellite in a maximum eclipse orbit.



(a) Optimal scheduling for scenario B case 2 – B2

(b) Optimal scheduling for scenario B case 3 – B3

(c) Energy analysis for scenario B case 2 – B2

(d) Energy analysis for scenario B case 3 – B3.

Source: **The author**.

It can be noticed that the payload activation times are reduced when compared with A2, where there was more energy available. In addition, all payloads enter a fully periodic operation regime. The power analysis for this case is presented in Figure 13c. Similarly to A2, the energy consumption follows closely the energy input and the battery access is more significant only at

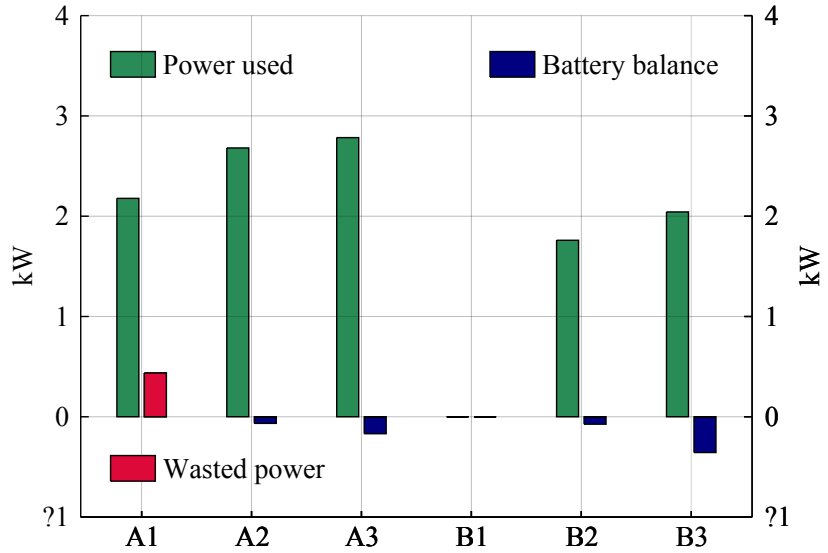
eclipse time. Furthermore, the state-of-charge is declining faster than in A2, an expected behavior given the longer eclipse time.

For case 3 (B3), the formulation is less restrictive with regards to battery access. A power footprint similar to that of case A3 is observed in Figure 13d, however the battery access is more intense with the SoC reaching the ρ limit at the end of the third orbit.

4.5.1.3 Results analysis

Figure 14 presents the power balance for the cases previously explored. Here power balance is defined as the total sum of power along the four orbits, so by "power used'" its meant all power consumed by the tasks along the 388 minutes (4 orbits), and by "battery balance", the net gain or net loss of power from the battery in the same period. Similarly, power wasted refers to the difference between power input and power utilized by tasks. Cases A1 and A2 can be directly compared since the parameters are all the same, except that A1 does not consider a battery. It is noticeable that the power consumption of the tasks is greater in A2 than in A1 and, in consequence, more work would be generated in orbit by scheduling tasks with the formulation proposed in this chapter. Despite consuming battery power, it is less than the energy wasted in A1's scheduling, showing the added mission value that the fuzzy battery formulation can deliver.

Figure 14 – Power balance in scenarios A and B.

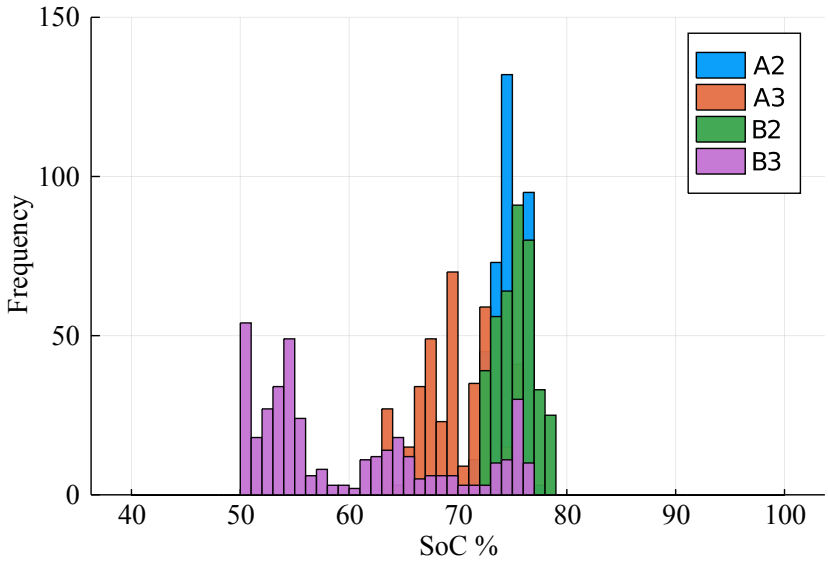


Source: **The author**.

Furthermore, the removal of constraints (25a) and (25b) in cases A3 and B3 results in more battery energy consumption and increased battery access, which decrease battery lifetime as shown in the histogram depicted in Figure 15. It can be noticed that there is a greater variation in SoC for the unrestricted cases, A3 and B3. Considering the average discharge rate of B3 (0.77 Ah), for instance, a battery such that used in FloripaSat-I mission would have its charge retention

capacity reduced to 60% in 67 days (SAMSUNG SDI CO., 2014), as opposed to 283 days for B2.

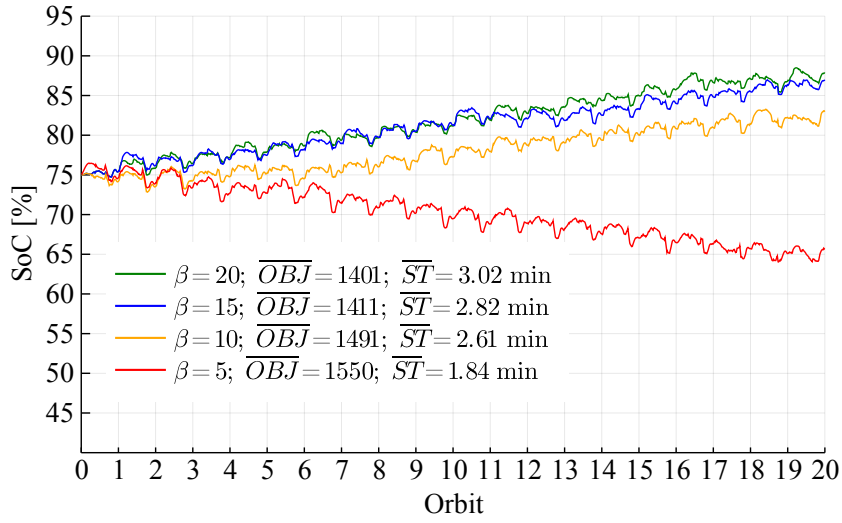
Figure 15 – SoC histogram.



Source: **The author**.

Finally, an analysis of the β value impact – the penalty weight for battery access in the objective function – is presented in Figure 16, where SoC is plotted for 20 consecutive orbits of decreasing eclipse time and four β values. Note that an increase in β means less battery usage and the average SoC will increase over time for $\beta > 10$. Notice that the solving time increases, and the objective value decreases, as the β value rises.

Figure 16 – SoC analysis for 20 consecutive orbits and different β values. $\overline{OBJ}$ is the average objective value per orbit and $\overline{ST}$ is the average solving time per orbit.



Source: **The author**.

4.5.2 Requisites: 6U mission

Here, the scheduling framework is applied to determine at which moments the payloads of a 6U nanosatellite can be activated, in a minute-by-minute basis, for four orbits. The scheduling data used as input was randomly generated except for the first 3 payloads, which were made periodic to demonstrate the effectiveness of the cross orbit coupling. The task data appear in Table 19, and the scheduling parameters are shown in Table 20, where T is the period of the chosen orbit (in minutes), whereas Q is a typical battery capacity for a 6U mission. The remaining parameters were left unchanged from the previous example, except for the initial SoC set to 85% and β that was set to zero.

Table 19 – Nanosatellite task data.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
u_{sj}	7	5	5	7	9	6	9	5	8	4	3	5	3	2	4	2	9	4
q_{sj}	0.19	0.45	0.11	0.19	70.19	0.11	0.28	0.28	0.02	1.11	1.11	0.85	0.72	0.85	1.11	0.85	0.85	1.11
$y_j^{\min}$	1	1	1	3	5	4	5	4	4	5	2	1	1	1	1	1	1	1
$y_j^{\max}$	8	8	9	62	69	21	47	63	49	18	19	6	3	3	4	2	4	2
$t_j^{\min}$	4	8	15	1	1	1	3	1	2	3	7	9	10	12	9	11	11	11
$t_j^{\max}$	4	8	15	6	6	19	19	9	11	18	38	47	20	28	25	51	45	72
$p_j^{\min}$	16	16	30	3	7	6	3	6	2	9	11	11	10	13	13	11	12	13
$p_j^{\max}$	16	16	30	29	22	35	28	40	29	34	65	58	68	88	44	56	53	65
$w_j^{\min}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$w_j^{\max}$	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

Source: **The author**.

Table 20 – Scheduling parameters.

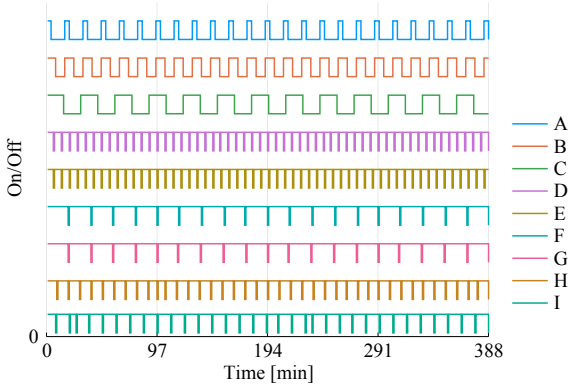
S	J	T	Q	V_b	e_c	e_d	SoC_0	ρ	γ	β
1	18	97	10 Ah	3.6 V	0.95	0.8	85%	0.5	5 A	0

Source: **The author**.

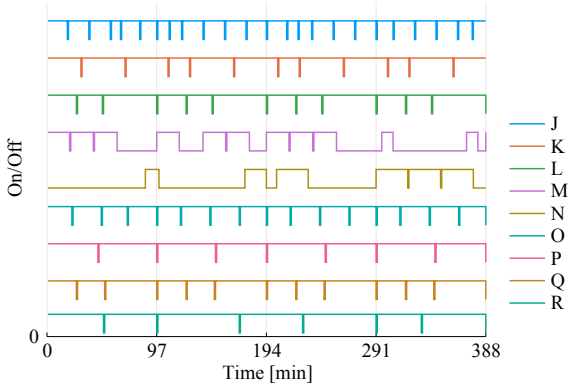
4.5.2.1 Scenario C – Max. Orbit

In this scenario, eighteen high energy impact tasks of a 6U mission (labeled A to R) are scheduled to show our methodology's versatility. The resulting schedule is presented in two parts: part one in Figure 17a which shows the schedule for tasks A to I; and part two in Figure 17b which depicts the schedule for tasks J to R. Notice that tasks A, B, and C have a strict period that remains consistent through all four orbits, demonstrating the cross orbit effectiveness of our methodology. The energy analysis, in Figure 17c, presents a result similar to cases A2 and B2, although here, the energy input is considerably larger given the size of the nanosatellite.

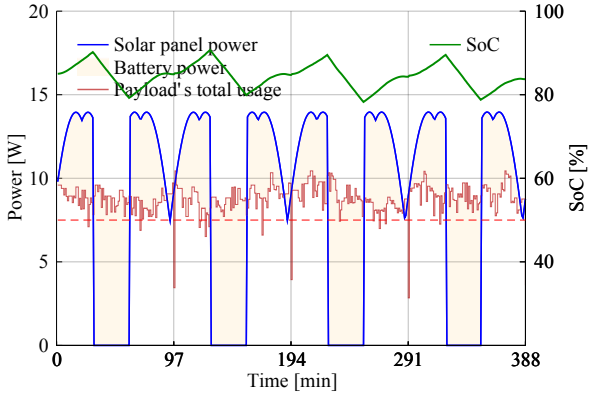
Figure 17 – Scheduling results for a 6U nanosatellite in a maximum eclipse orbit.



(a) Optimal scheduling for scenario C – jobs A to I.



(b) Optimal scheduling for scenario C – jobs J to R.



(c) Energy analysis for scenario C.

Source: **The author**.

4.6 DISCUSSION

In conclusion, the work presented in this chapter has improved the state-of-the-art in nanosatellite scheduling by: incorporating a realistic battery model along with strategies for battery lifetime extension; providing a novel methodology for multi-orbit scheduling; and improving upon an existing methodology for multi-orbit power input calculation.

These methodologies were fully demonstrated and characterized by investigating three different scheduling scenarios. From the results, one can infer that sometimes the formulation including the battery is the only one capable of making the model feasible, mainly because of the moments of eclipse encountered in the orbit. Furthermore, the formulation better manages the available energy in orbit, maximizing the usefulness of the satellite, relative to the methodology presented in Chapter 3.

Nevertheless, solving the MILP formulation of the scheduling problem for a larger time set $\mathcal{T}$ in a reasonable time would allow obtaining a task plan second by second instead of minute by minute and, therefore, a solution including all module tasks of the satellite, not only the payload activation tasks. A higher number of tasks with more granular power footprints could result in an even better power consumption match with the available power, leading to further

mission efficacy. The next chapter addresses this research path.

5 BRANCH-AND-PRICE FORMULATION

Notwithstanding the contributions to the state-of-the-art presented in the previous chapter, the formulation becomes impractical to solve large instances, when the computational time for scheduling one orbit can take longer than the orbit period itself. A nanosatellite scheduling problem can be made larger by including more tasks or increasing orbit duration, the former being particularly desirable for planning missions of larger stacking where proportionally more work has to be managed. Furthermore, more tasks with finer power footprints can lead to better power management and further mission efficacy.

Pursuing this research path, a decomposition and solution strategy is formalized so that it can cope with larger sets of tasks and an extended time horizon, within a reasonable computation time. More specifically, the mixed-integer linear programming (MILP) formulation is decomposed by tasks according with the Dantzig-Wolfe decomposition – a technique that has already been applied to earth observation satellites, as can be seen in Hu et al. (2019) and Wang, Demeulemeester, Hu, and Wu (2020), – resulting in a profile-based formulation that has never been explored in the literature. In this novel formulation, the main decision variables are related to feasible task profiles, such that each profile indicates the active times of its corresponding task. To solve this formulation, a branch-and-price (B&P) algorithm is created, such that enables the effective planning of more complex missions, in an optimal or nearly-optimal manner and within practicable computation times.

Nevertheless, the formulation presented in Chapter 4 is modified to remove the battery charge and discharge efficiency, introducing only one efficiency constant, and the battery usage penalty is removed from the objective formulation, for simplicity purposes, along with the subsystem index, such that the model considered here is as follows:

$$\max \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} u_j x_{jt}, \tag{35a}$$

$$\text{s.t.} \sum_{j\in\mathcal{J}} q_j x_{jt} \leq r_t + \gamma\ V_b\ (1-\alpha_t), \qquad \forall t \in \mathcal{T}, \tag{35b}$$

$$b_t = r_t\ - \sum_{j\in\mathcal{J}} q_j x_{jt}, \qquad \forall t \in \mathcal{T}, \tag{35c}$$

$$i_t = \frac{b_t}{V_b}, \qquad \forall t \in \mathcal{T}, \tag{35d}$$

$$\mathrm{SoC}_{t+1} = \mathrm{SoC}_t + \frac{i_t\ e}{60\ Q}, \qquad \forall t \in \mathcal{T}, \tag{35e}$$

$$\mathrm{SoC}_t \leq 1, \qquad \forall t \in \mathcal{T}, \tag{35f}$$

$$\mathrm{SoC}_t \geq \rho, \qquad \forall t \in \mathcal{T}, \tag{35g}$$

$$\mathrm{SoC}_T \leq \mathrm{SoC}_1 (1 + \Delta^{\mathrm{SoC}}), \tag{35h}$$

$$\mathrm{SoC}_T \geq \mathrm{SoC}_1 (1 - \Delta^{\mathrm{SoC}}), \tag{35i}$$

$$\phi_{j1} \geq x_{j1}, \qquad \forall j \in \mathcal{J}, \tag{35j}$$

$$\phi_{jt} \geq x_{jt} - x_{j(t-1)}, \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T} \setminus \{1\}, \tag{35k}$$

$$\phi_{jt} \leq x_{jt}, \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}, \tag{35l}$$

$$\phi_{jt} \leq 2 - x_{jt} - x_{j(t-1)}, \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T} \setminus \{1\}, \tag{35m}$$

$$\sum_{t \in \mathcal{T}} \phi_{jt} \geq y_j^{\min}, \qquad \forall j \in \mathcal{J}, \tag{35n}$$

$$\sum_{t \in \mathcal{T}} \phi_{jt} \leq y_j^{\max}, \qquad \forall j \in \mathcal{J}, \tag{35o}$$

$$\sum_{l=t}^{t+t_j^{\min}-1} x_{jl} \geq t_j^{\min} \phi_{jt}, \qquad \forall t \in \left\{1, \ldots, T - t_j^{\min} + 1\right\}, \forall j \in \mathcal{J}, \tag{35p}$$

$$\sum_{l=t}^{t+t_j^{\max}} x_{jl} \leq t_j^{\max}, \qquad \forall t \in \left\{1, \ldots, T - t_j^{\max}\right\}, \forall j \in \mathcal{J}, \tag{35q}$$

$$\sum_{l=t}^{T} x_{jl} \geq (T - t + 1)\phi_{jt}, \qquad \forall t \in \left\{T - t_j^{\min} + 2, \ldots, T\right\}, \forall j \in \mathcal{J}, \tag{35r}$$

$$\sum_{l=t}^{t+p_j^{\min}-1} \phi_{jl} \leq 1, \qquad \forall t \in \left\{1 + w_j^{\min}, \ldots, w_j^{\max} - p_j^{\min}\right\}, \forall j \in \mathcal{J}, \tag{35s}$$

$$\sum_{l=t}^{t+p_j^{\max}-1} \phi_{jl} \geq 1, \qquad \forall t \in \left\{1 + w_j^{\min}, \ldots, w_j^{\max} - p_j^{\max}\right\}, \forall j \in \mathcal{J}, \tag{35t}$$

$$x_{jt} = 0, \qquad t = 1, \ldots, w_j^{\min}, \forall j \in \mathcal{J}, \tag{35u}$$

$$x_{jt} = 0, \qquad t = w_j^{\max}, \ldots, T, \forall j \in \mathcal{J}, \tag{35v}$$

$$x_{jt} \in \{0, 1\}, \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}, \tag{35w}$$

$$\phi_{jt} \in \{0, 1\}, \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}, \tag{35x}$$

$$0 \leq \alpha_t \leq 1, \qquad \forall t \in \mathcal{T}, \tag{35y}$$

$$b_t, i_t \in \mathbb{R}, \qquad \forall t \in \mathcal{T}. \tag{35z}$$

The text of this chapter was published as an article in Rigo, Seman, Camponogara, Morsch Filho, Bezerra, and Munari (2022).

5.1 DANTZIG-WOLFE DECOMPOSITION AND COLUMN GENERATION ALGORITHM

The MILP formulation presented in the previous chapter can become intractable by general-purpose optimization solvers when big instances are considered – finer time granularity or missions above 3U size, with several payloads. A more effective approach can be designed by exploring the special structure of the coefficient matrix of this formulation. To this end, the Dantzig-Wolfe decomposition (DANTZIG; WOLFE, 1960) is implemented here, which enables the use of the column generation technique to improve tractability. This decomposition divides the original formulation into a master problem and one or more subproblems. The master

problem considers only a subset of the constraints in the original formulation, and is defined by variables related to integer points and rays of the polyhedra related to the constraints that were not included. The advantage is that these points and rays can be generated iteratively, by resorting to the subproblem(s), in a framework known as the column generation technique (LÜBBECKE; DESROSIERS, 2005; VANDERBECK; WOLSEY, 2010; GONDZIO; GONZÁLEZ-BREVIS; MUNARI, 2013). The following subsections formalize the decomposition of the MILP formulation and describe the column generation scheme. The additional nomenclature used hereafter is introduced in Table 21.

Table 21 – Added sets, indexes, variables and constants.

Notation	Definition
Sets	
$\mathcal{K}_j$	set of feasible profiles of task $j \in \mathcal{J}$.
$\mathcal{K}_j^*$	a subset of feasible profiles of task $j \in \mathcal{J}$.
Variables	
$\xi_j^k \in \{0,1\}$	takes the value of 1 if, and only if, a profile $k \in \mathcal{K}_j$ is selected.
Parameters	
$\theta_{jt}^k \in \{0,1\}$	is equal to 1 if, and only if, task j is in execution at time $t \in \mathcal{T}$ according with profile $k \in \mathcal{K}_j$.
$c_j^k > 0$	total priority associated to the k^{th} profile of task j.
$q_j^k(t) > 0$	power usage associated to the k^{th} profile of task j at time t.

Source: **The author**.

5.1.1 Master problem

Consider the MILP model (35a)–(35z) and let $\mathcal{K}_j$ be the set of all feasible schedules for task j, *i.e.*, schedules that satisfy constraints (35j)– (35x). Each schedule is represented by a profile vector $\theta_j^k \in \{0,1\}^T$ such that task j is active at time t in the k^{th} profile if, and only if, $\theta_{jt}^k = 1$. Let ξ_j^k be the binary variable that assumes the value of 1 if, and only if, the k^{th} profile is selected for task j. Furthermore, let $c_j^k = u_j \sum_{t=1}^{T} \theta_{jt}^k$ be the total priority associated to the k^{th} profile of task j, and $q_j^k(t) = q_j \theta_{jt}^k$ be the power usage associated with this task at time t.

For each task $j \in \mathcal{J}$, we have the following relationship between variables x_{jt} and ξ_j^k:

$$x_{jt} = \sum_{k \in \mathcal{K}_j} \theta_{jt}^k \xi_j^k, \ \forall t \in \mathcal{T}, \text{ with } \sum_{k \in \mathcal{K}_j} \xi_j^k = 1. \tag{36}$$

Using this equivalence and applying the Dantzig-Wolfe decomposition to the MILP formulation (35a)–(35z), we obtain the following Master Problem (MP):

$$MP: \max \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}_j} c_j^k \, \xi_j^k, \tag{37a}$$

$$\text{s.t.} \sum_{k \in \mathcal{K}_j} \xi_j^k = 1, \qquad \forall j \in \mathcal{J}, \tag{37b}$$

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}_j} q_j^k(t) \xi_j^k + \gamma\ V_b\ \alpha_t \leq r_t + \gamma\ V_b, \qquad \forall t \in \mathcal{T}, \tag{37c}$$

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}_j} q_j^k(t) \xi_j^k + b_t = r_t, \qquad \forall t \in \mathcal{T}, \tag{37d}$$

$$\text{(35d)-(35i)}, \text{(35y)}, \text{(35z)},$$

$$\xi_j^k \in \{0,1\}, \qquad \forall j \in \mathcal{J},\ \forall k \in \mathcal{K}_j. \tag{37e}$$

The Master Problem (MP) consists of finding a profile for each task such that constraints (37b) to (37e) are satisfied, while maximizing the mission value (37a). Notice that constraints (37b) state that one, and only one, profile must be selected for each task j. Constraints (37c) limit the energy consumption to the available capacity at each time t of the orbit. Constraints (37d) and (35d)-(35i) refer to the battery energy management. Finally, constraints (35y), (35z) and (37e) define the domain of the decision variables.

5.1.2 Column generation

The number of feasible profiles in the MP is combinatorial, rendering it impractical to enumerate all profiles θ_j^k. Hence, the CG technique is used to solve its LP relaxation. This technique allows us to consider subsets $\mathcal{K}_j^* \subseteq \mathcal{K}_j$ and iteratively expand them until an optimal solution of the LP relaxation is ensured. Hence, in each iteration of the method, the following RMP is solved:

$$RMP: \max \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}_j^*} c_j^k\ \xi_j^k, \tag{38a}$$

$$\text{s.t.} \sum_{k \in \mathcal{K}_j^*} \xi_j^k = 1, \qquad \forall j \in \mathcal{J}, \tag{38b}$$

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}_j^*} q_j^k(t)\ \xi_j^k + \gamma\ V_b\ \alpha_t \leq r_t + \gamma\ V_b, \qquad \forall t \in \mathcal{T}, \tag{38c}$$

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}_j^*} q_j^k(t)\ \xi_j^k + b_t = r_t, \qquad \forall t \in \mathcal{T}, \tag{38d}$$

$$\text{(35d)-(35i)}, \text{(35y)}, \text{(35z)},$$

$$\xi_j^k \geq 0, \qquad \forall j \in \mathcal{J},\ \forall k \in \mathcal{K}_j^*. \tag{38e}$$

Let μ_j, π_t and ν_t be the dual variables associated with the constraints (38b)-(38d), respectively. Given a corresponding optimal dual solution $\bar{\mu}_j$, $\bar{\pi}_t$ and $\bar{\nu}_t$ of the RMP, it is verified if new variables with positive reduced cost can be generated for the RMP resorting to the following pricing subproblems PS_j, for each task $j \in \mathcal{J}$:

$$PS_j:\ \tilde{c}_j = \max_{k \in \mathcal{K}_j \setminus \mathcal{K}_j^*} c_j^k\ -\ \bar{\mu}_j - \sum_{t=1}^{T} (\bar{\pi}_t + \bar{\nu}_t) q_j^k(t), \tag{39}$$

where $\tilde{c}_j$ is the largest reduced cost that can be achieved by a profile of task j. Solving the pricing problem PS_j in the form (39) is not practical, since it amounts to enumerating all columns that correspond to feasible task schedules. Instead, we can solve an optimization problem that generates a profile in $\mathcal{K}_j \backslash \mathcal{K}_j^*$, based on the constraints that define a feasible schedule, as follows:

$$PS_j: \ \tilde{c}_j = \max_{x_j, \phi_j} u_j \sum_{t=1}^{T} x_{jt} - \bar{\mu}_j - q_j \sum_{t=1}^{T} (\bar{\pi}_t + \bar{\nu}_t)\, x_{jt}, \tag{40a}$$

$$\text{s.t. (35j)-(35x) [Restricted to task } j\text{]}, \tag{40b}$$

which is a 0-1 integer program on the variables $x_j = (x_{jt} : \forall t)$ and $\phi_j = (\phi_{jt} : \forall t)$.

An optimal solution of the RMP is optimal for the LP relaxation of the MP if

$$\max \left\{ \tilde{c}_j : j \in \mathcal{J} \right\} \leq 0, \tag{41}$$

meaning that there are no columns corresponding to profiles in $\mathcal{K}_j \backslash \mathcal{K}_j^*$ that have a positive reduced cost and, hence, we can halt the column generation algorithm.

5.2 BRANCH-AND-PRICE

If the optimal solution obtained at the end of the CG algorithm is integer, then it is optimal for the MP as well. Otherwise, we need to embed the CG algorithm in a branch-and-bound framework to find the integer optimal solution, a procedure that is known as branch-and-price (B&P) (BARNHART et al., 1998; MUNARI; GONDZIO, 2013). To propose a B&P algorithm for solving the MP (37), in what follows an effective branching scheme is presented and the changes that are required in the RMP and subpproblems of the CG algorithm are shown. Additionally, a MIP heuristic that helps the method finding integer feasible solutions using the generated profiles is presented.

5.2.1 Branching scheme

The branching scheme is based on the original variables x_{jt} of the MILP formulation, using the equivalence (36), and the branching constraints are imposed explicitly in the RMP, in terms of variables ξ_j^k. At each node l, after finishing the column generation procedure and certifying that the node cannot be pruned, it is selected the pair of indices (j,t) such that x_{jt} is the most fractional – in case of ties its randomly selected – using the following expression:

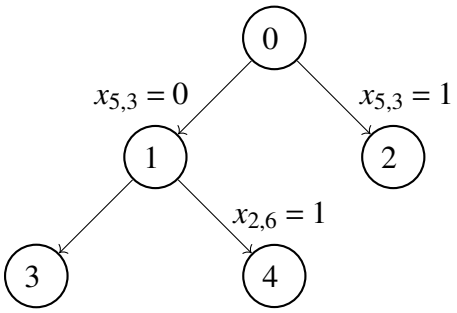
$$\underset{\forall (j,t)}{\operatorname{argmin}} \ \left| 0.5 - \sum_{k \in \mathcal{K}_j^*} \theta_{jt}^k \xi_j^k \right|. \tag{42}$$

Then, if x_{jt} is fractional for the selected pair (j,t), two branches are created imposing $x_{jt} = 0$ in one child node and $x_{jt} = 1$ in the other as follows, respectively:

$$\sum_{k \in \mathcal{K}_j^*} \theta_{jt}^k \xi_j^k = 0 \quad \text{and} \quad \sum_{k \in \mathcal{K}_j^*} \theta_{jt}^k \xi_j^k = 1.$$

Its denoted as $\mathcal{L}_0^{(l)} = \{(j,t) : x_{jt} = 0 \text{ at node } l\}$ and $\mathcal{L}_1^{(l)} = \{(j,t) : x_{jt} = 1 \text{ at node } l\}$ the sets of indices related to the branching constraints imposed at node l, all the way from the root node. To illustrate the definition of these sets, an example of a branching tree is shown in Figure 18, in which for node 4 we have the sets $\mathcal{L}_0^{(4)} = \{(5,3)\}$ and $\mathcal{L}_1^{(4)} = \{(2,6)\}$, as the branching constraints $x_{53} = 0$ and $x_{26} = 1$ are both enforced in this node.

Figure 18 – Illustration of branching tree.



Source: **The author**.

5.2.2 Changes to RMP and subproblems

Let $RMP^{(l)}$ be the restricted master problem of a given node l, which is defined as follows:

$$RMP^{(l)}: \max \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}_j^*} c_j^k \, \xi_j^k, \tag{43a}$$

$$\text{s.t. (38b)-(38d), (35d)-(35i), (35y), (35z)}$$

$$\sum_{k\in\mathcal{K}_j^*} \theta_{jt}^k \xi_j^k = 0, \qquad \forall\,(j,t) \in \mathcal{L}_0^{(l)}, \tag{43b}$$

$$\sum_{k\in\mathcal{K}_j^*} \theta_{jt}^k \xi_j^k = 1, \qquad \forall\,(j,t) \in \mathcal{L}_1^{(l)}, \tag{43c}$$

$$\xi_j^k \geq 0, \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}_j^*. \tag{43d}$$

Notice that constraints (43b) and (43c) enforce the branching constraints at each $RMP^{(l)}$. Let $\zeta_{j,0}^t$ and $\zeta_{j,1}^t$ be dual variables associated to these branching constraints, respectively. The objective function of each pricing subproblem needs to change to take into account these dual variables. For each task j, the subproblem becomes:

$$PS_j: \widetilde{c}_j = \max u_j \sum_{t=1}^{T} x_{jt} - \mu_j - q_j \sum_{t=1}^{T} (\pi_t + \nu_t)\, x_{jt} - \sum_{(j,t)\in\mathcal{L}_0^{(l)}} x_{jt}\zeta_{j,0}^t - \sum_{(j,t)\in\mathcal{L}_1^{(l)}} x_{jt}\zeta_{j,1}^t, \tag{44a}$$

$$\text{s.t. : (35j)-(35x) [Restricted to task } j\text{].} \tag{44b}$$

5.2.3 MIP heuristic

Finally, a simple MIP heuristic is applied to increase the chances of obtaining good feasible solutions that can be used to improve the lower bound of the B&P method (SADYKOV et al., 2019; ALVAREZ; MUNARI, 2017). This heuristic is called at the end of each node, right before branching, and consists of solving a compact MIP model based on the RMP, using a general-purpose MIP solver. The idea is that the MIP solver may determine an integer feasible solution based on the combination of the profiles (columns) generated so far, which may be better than the incumbent solution.

The first step of the heuristic is to redefine as binary the domain of all variables ξ_j^k in the last RMP solved in the node, resulting in the problem named as $HRMP^{(l)}$ for node l. Notice that this is the same problem as the $RMP^{(l)}$ we have at the end of the column generation procedure, except that it has the additional constraints $\xi_j^k \in \{0,1\}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}_j^*$. The $HRMP^{(l)}$ is solved as a compact model by a general-purpose MIP solver (and without generating any further columns during this process). Any feasible solution of this problem is clearly feasible for the MP. Therefore, if a feasible solution with objective value larger than the current Lower Bound (LB) is obtained, it replaces the incumbent and its value becomes the new LB.

5.2.4 Branch-and-price procedure

Each node of the search tree involves a call to the CG algorithm, specific lower/upper bound updates, node cut-off criteria and the branching strategy. For $RMP^{(1)}$, the first RMP, its provided an initial set of feasible columns so that the proper duals are generated to set the pricing subproblems for the first time. In this implementation, is created a column for each task where all values are 1's and modified the first RMP by including in the objective function a highly negative term multiplying the cost of each of these artificial columns.

Algorithm 2 presents the pseudo-code of the routine performed at each node in the search tree. The function `Node` receives as parameter the node number (l), the lower bound (LB) and the upper bound (UB). Let $\sigma = (\bar{\mu}_j, \bar{\pi}_t, \bar{\nu}_t, \bar{\zeta}_{j,0}^k, \bar{\zeta}_{j,1}^k : \forall j \in \mathcal{J}, \forall t \in \mathcal{T})$ be the relevant components of the optimal dual solution of (43). Then, the first step is to run the CG algorithm, shown in lines 2 to 10. Furthermore, the pricing problem was solved in parallel, simultaneously for each task using the general-purpose branch-and-cut of the Gurobi solver (GUROBI OPTIMIZATION, 2016) and returning multiple columns. After running the column generation loop, the node is checked for pruning based on the LB (lines 11 and 12), and then the upper bound is updated, if necessary (lines 13 and 14).

The next step is to run the MIP heuristic based on the $HRMP^{(l)}$ (line 15) and then update the lower bound depending on the solution returned by this heuristic (lines 17 to 19). If the objective value $Z_{H^{(l)}}$ of the returned solution is equal to the upper bound, then it closes the integrality gap. Therefore, an optimal integer solution has been found and the method terminates successfully (lines 20 and 21). Finally, after updating the gap value (line 22), the procedure

Algorithm 2: Solving a node in the branch-and-price algorithm

```
1  Function Node(l, LB, UB^(l)):
2      do
3          Solve RMP^(l) to obtain its optimal value Z_RMP^(l) and the respective dual
             variables σ ;
4          if RMP^(l) is infeasible then
5              return ∅; // Prune node
6          for each j ∈ J do
7              Set PS_j using σ and solve it ;
8              if c̃_j > 0 then
9                  Add the optimal schedule of PS_j to K_j^(l);
10     while c̃_j > 0  for at least one j ∈ J;
11     if ⌊Z_RMP^(l)⌋ ≤ LB then
12         return ∅; // Prune node
13     if Z_RMP^(l) < UB^(l) then
14         UB^(l) = ⌊Z_RMP^(l)⌋; // Update upper bound
15     Solve HRMP^(l); // MIP heuristic
16     if HRMP^(l) returns a feasible solution then
17         Let Z_H^(l) be the value of the returned solution;
18         if Z_H^(l) > LB then
19             LB = Z_H^(l); // Update lower bound (new incumbent solution)
20         if Z_H^(l) = UB^(l) then
21             return 1; // Integer optimal solution found!
22     Let (j, t) be the pair of indices obtained using (42);
23     Compute x_jt using the equivalence in (36);
24     if x_jt is fractional then
25         Define a new node using the branching constraints L_0^(l) ∪ {(j, t)} and insert it
             into the list of nodes;
26         Define a new node using the branching constraints L_1^(l) ∪ {(j, t)} and insert it
             into the list of nodes;
27     else
28         if Z_RMP^(l) > LB then
29             LB = Z_RMP^(l); // Update lower bound (new incumbent
                   solution)
```

applies the branching scheme described in Section 5.2.1 (lines 23 and 24). If the solution in terms of the original variables is fractional, two child nodes are created and added to the list of nodes (lines 25 to 27), to be processed later in the search tree.

For each branching candidate variable found in the variable selection strategy – Equation (42) – the node function will be called passing the current variable to be constrained, along with the previous variables all the way back to the top of the search tree. Clearly, the strategy

here is depth-first search – a common strategy for scheduling problems (FRISKE; BURIOL; CAMPONOGARA, 2018; MAENHOUT; VANHOUCKE, 2010) – since the left branch will always be explored first, and the right will only be called if the left returns without a solution. If both return without a solution, then the next candidate for branching in the node will be explored, and so on.

5.3 COMPUTATIONAL EXPERIMENTS

In this section, is reported the results of computational experiments carried out to verify the performance of the proposed B&P method[1]. Firstly, is presented the attitude and orbital parameters used for the power input calculation evaluation the B&P algorithm. Then the B&P procedure is compared to traditional solution approaches, and lastly, illustrative but realistic schedules are presented.

Its considered here that tasks refer to the activation and deactivation time of payloads, which are high resource impact components of nanosatellites and require careful management given their critical importance for mission success. Therefore, in this problem, tasks are hardware modules integrated into the spacecraft, and their quantity depends on the nanosatellite size (i.e., 1U, 2U, etc.), and does not change during the mission. These high energy impact tasks have to be pre-scheduled on Earth (i.e., the ground station), offline, and later sent to the nanosatellite, which will then take leverage of this information to define the order of tasks to be performed.

The methodologies presented in Section 5.2 were implemented in the Julia programming language using the JuMP library (DUNNING; HUCHETTE; LUBIN, 2017) and solved using the Gurobi solver in a server with an Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz processor (20 cores, 40 threads), 64 GB of RAM, and Ubuntu 20.04.2 LTS 64 bits.

5.3.1 Power input calculation

The input data used by the methodology of Filho et al. (2020) for power harvesting calculation was based on the FloripaSat-I mission, with an altitude of 628 km and an orbit duration of 97 minutes (MARCELINO et al., 2020). Notice that when $T > 97$ the schedule is solved for more than one orbit period at once, since the time granularity explored here will be minutes. The orbital parameters are presented in Table 22, where: i is the orbit inclination [°]; Ω is the right ascension of the ascending node [°]; e is eccentricity [-]; ω is the argument of perigee [°]; M is mean anomaly [°]; and n is mean motion [rev/day].

Two attitudes considering an orbit of half-maximum eclipse and stacking of 3U, and one with maximum eclipse and a stacking 6U, will be explored to elucidate the spacecraft's kinematics and size impact on power calculation. The obtained power vector is multiplied by an EPS efficiency of 0.85, finally resulting in the power available for the tasks – the input vector for the scheduling methodology.

[1] The instances are available at https://github.com/c-a-rigo/nanosatellite-scheduling-instances.

Table 22 – Orbital parameters.

Nanosatellite	i	Ω	e	ω	M	n
FloripaSat-I	97.95	225.78	0.0016	111.38	248.91	14.82

Source: **The author**.

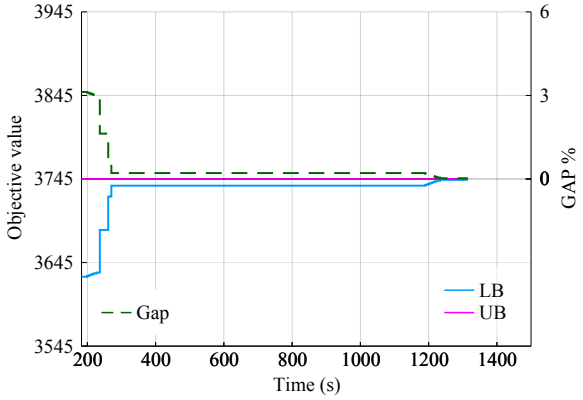
5.3.2 Initial evaluation

For initial analysis of the B&P algorithm behavior, a case with randomly generated scheduling data, $T = 194$ and $J = 9$, was solved. Notice that, because $T = 194$, the result will be a plan for two consecutive orbit periods – although the decision maker could as easily schedule one orbit in a time resolution of 30 seconds. Two versions of the B&P algorithm are run: in the first, it generates at most one column per each call to a subproblem PS_j, whereas in the second, it can generate up to 30 columns per call (corresponding to feasible solutions that are found and stored by the solver Gurobi while solving PS_j). Figure 20a shows the evolution of bounds resulting from the column generation procedure, with the pricing subproblem returning at most one column per call to each PS_j.

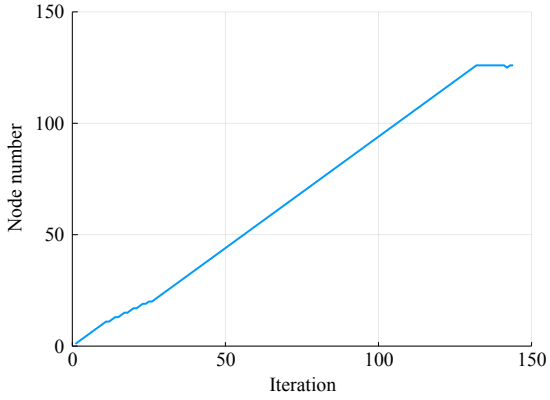
The optimal value obtained by the CG algorithm at the root node is $Z_{RMP^{(0)}} = 3745.736$ and by rounding down this value we obtain the upper bound 3745, which coincides with the optimal value of the instance ($Z_{optimal}$). The solution is still fractional though, and the lower bound obtained by the MIP heuristic is relatively low – with an initial $GAP = 3.124\%$. This lower bound rises closer to $Z_{optimal}$ after a few iterations, as the search tree deepens. Eventually, the lower and upper bound converge, and the optimal solution is found. Figure 19b shows in which node the algorithm was at each iteration. It can be noticed that, for the particular instance illustrated in Figure 19b, node pruning was only applied at the beginning and at the end of the search tree; in this sense, it was not possible to observe a direct correlation between iteration and node number (tree depth).

Figure 20b shows the bounds evolution at each iteration when solving the same instance but with the generation of several columns per call to each sub-problem PS_j, corresponding to the k best columns found for each task. The result is similar to the previous case, with the upper bound reaching the optimal value in the first node, although now the lower bound rises much faster. One can notice that finding the optimal solution takes about the same computation time with single or multiple column returns. Nevertheless, the GAP starts so low (0.053%) that an almost optimal solution is available in the first node, which could be a sufficiently good solution for the decision-maker, obtained with far less computational effort in comparison to the single-column case.

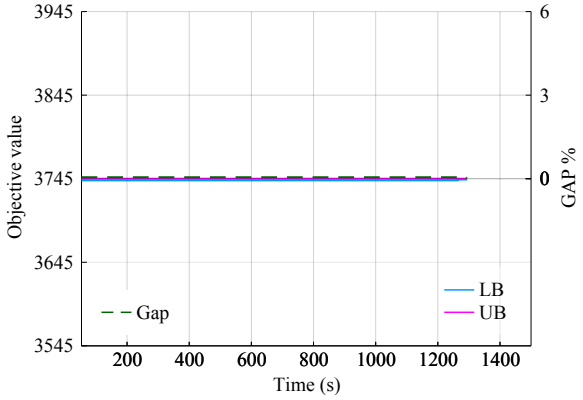
Figure 19 – Branch-and-price bounds and node processing evolution in the search tree with single and with multiple columns.



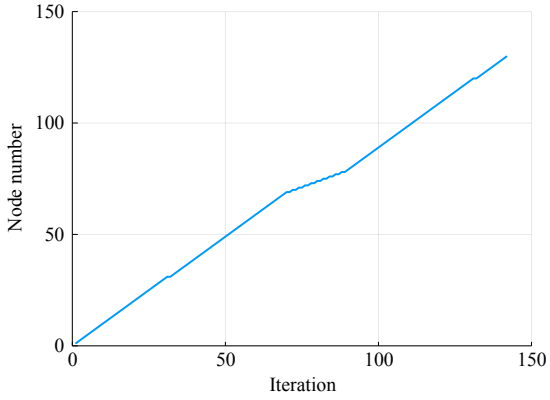
(a) Bounds returning a single column. Solve time = 1312 s.



(b) Search tree for the single-column case.



(c) Bounds returning multiple columns. Solve time = 1291 s.



(d) Search tree for the multiple-column case.

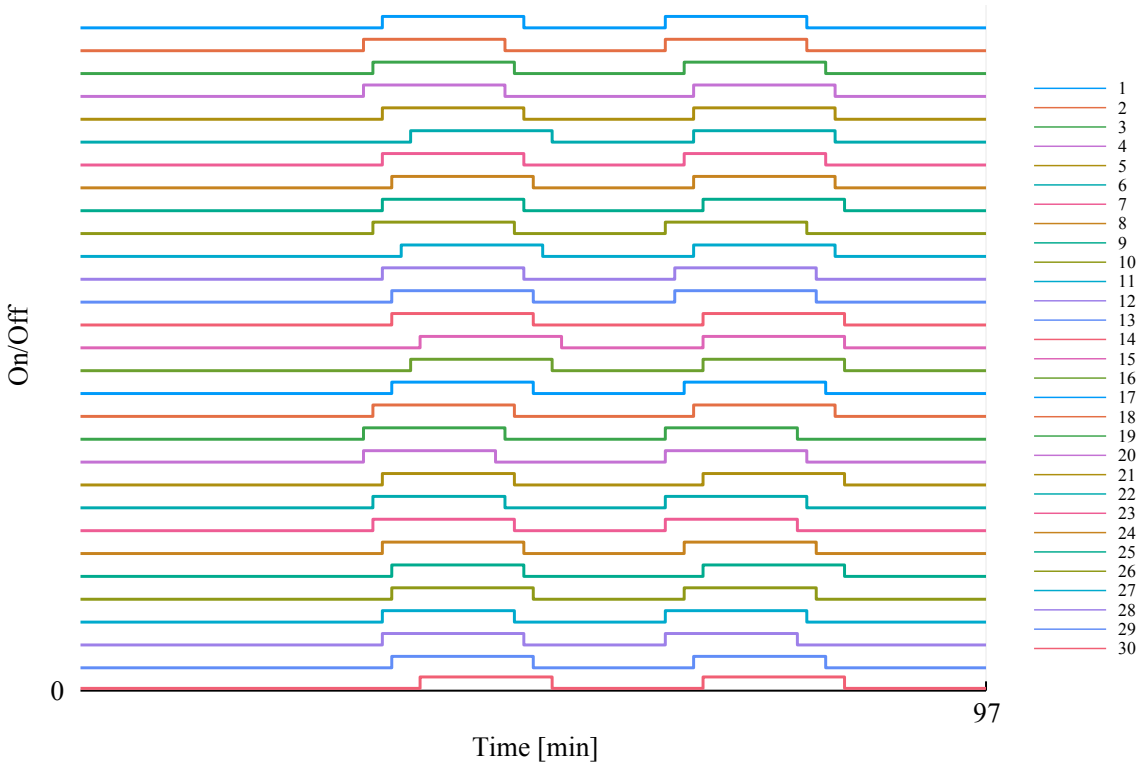
Source: **The author**.

This behavior was observed in all instances considered by the computational analysis. In fact, the tests showed that in instances where an optimal solution was found deep in the tree for the single-column case, the multiple-column version proved optimality in the first node with relatively little computation time. This is the case for all instances of Table 24 where the solution was found in the first node.

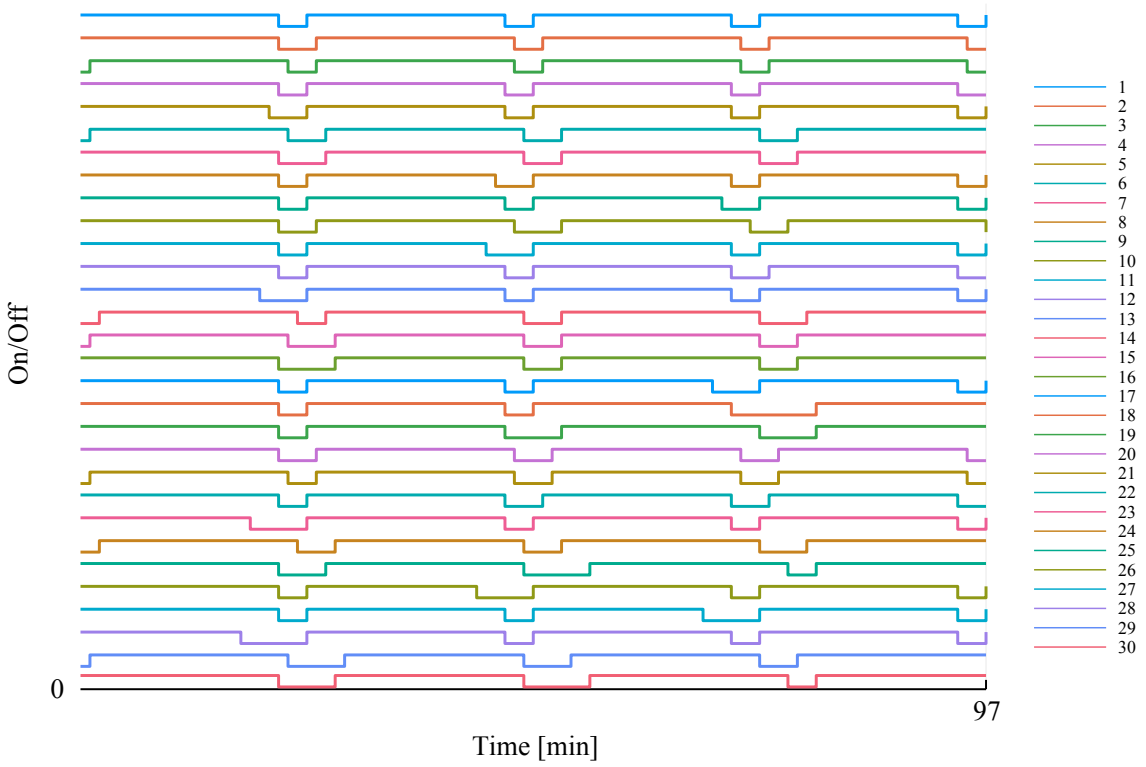
5.3.2.1 Column generation

To investigate the causes of the bound behavior previously reported, we explore the first 30 columns returned from the pricing of each job in a smaller example, with $T = 97$ and $J = 9$. Figure 20 presents these columns for two jobs, 1 and 7, respectively.

Figure 20 – First 30 columns returned from the first pricing procedure.



(a) Job 1 columns. Task parameters: $y^{\min}$=2; $y^{\max}$=4; $t^{\min}$=10; $t^{\max}$=15; $p^{\min}$=30; $p^{\max}$=78; $w^{\min}$=30; $w^{\max}$=81.



(b) Job 7 columns. Task parameters: $y^{\min}$=1; $y^{\max}$=3; $t^{\min}$=2; $t^{\max}$=7; $p^{\min}$=21; $p^{\max}$=24; $w^{\min}$=0; $w^{\max} = T$.

Source: **The author**.

These plots show that each column is unique, and the variation between them is slight. This indicates that the pricing generates unique but similar columns, characterizing a restricted problem. In fact, if we analyze the standard deviation of the 30 columns for each job, presented in Table 23, it can be seen that this is indeed the case, where only job 7 has a standard deviation above 1.

Table 23 – Standard deviation between columns for each job.

Job	1	2	3	4	5	6	7	8	9
σ	0.49	0.80	0.87	0.50	0.76	0.98	1.06	0.66	0.57

Source: **The author**.

Therefore, given that in the first node the pricing procedures will generate quite a significant amount of columns, we can assign the ability of the $RMP^{(1)}$ to reach optimal bounds– and in many cases, the optimality itself – to this sufficient enough diversity and of unique columns.

5.3.3 Comparison with the traditional approach

Table 24 presents the results for the multiple-column B&P algorithm in comparison with solving the MILP formulation (35a)–(35z) using the general-purpose solver of Gurobi, for 42 randomly generated instances wherein the cardinality of $\mathcal{T}$ varies from 97 to 291 and for cases with 9, 13, 18, 20, 22 and 24 tasks. For each instance, the table shows the instance number (#), the number of time periods (T), and the number of tasks (J). Then, for each solution approach, the table reports the objective value of the best feasible solution obtained (Obj value), which is the optimal value if the total computation time is less than 7200 seconds; the best upper bound obtained by the approach (Best UB); the computation time to solve the instance (CPU); the relative optimality gap, computed as GAP = (Best UB – Obj value)/(Best UB + $1e^{-20}$); and the final status of the approach (End Cause), where *Optimal* means that the instance was solved to proven optimality, while *Time* means that the method finished after reaching the time limit. For the B&P method, there is one additional column, showing the number of nodes processed in the search tree (Nodes). The best results are marked with a star, indicating the solution strategy that achieved the best time or GAP within the time limit of 7200 seconds.

The results in Table 24 show that the B&P algorithm performs better in solving the nanosatellite scheduling problem when compared to the commercial alternative. In 29 out of 42 instances, the proposed algorithm achieved optimality with the best computation time, on average, 88.58% faster than Gurobi. In instance 16, for example, the B&P strategy solved the problem to optimality in 21 seconds, at the root node. In contrast, Gurobi used all available time and could not prove optimality. The complexity increase with problem size is also noticeable for both solution approaches, as it becomes more costly to solve instances of $T = 194$ or larger and with 9/13 tasks, and far more costly for $T = 154$ or higher and with 18 tasks.

Table 24 – Performance of the multiple-column B&P algorithm *vs.* MILP formulation using the general-purpose solver Gurobi for 21 instances (time limit = 7200 seconds).

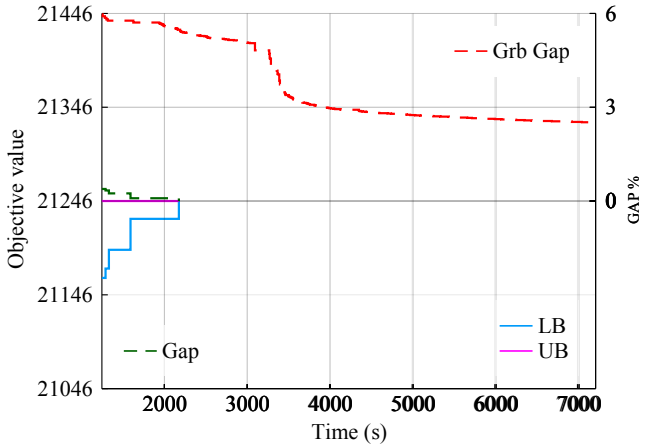
			B&P						**Gurobi**				
#	T	J	**Obj value**	**Best UB**	**CPU (s)**	**Gap**	**Nodes**	**End cause**	**Obj value**	**Best UB**	**CPU (s)**	**Gap**	**End cause**
1	97	9	1141	1141	⋆8	0.0000%	1	Optimal	1141	1141	13	0.00%	Optimal
2	120	9	2284	2284	⋆16	0.0000%	2	Optimal	2284	2284	128	0.00%	Optimal
3	125	9	2553	2553	15	0.0000%	2	Optimal	2553	2553	⋆2	0.00%	Optimal
4	154	9	2765	2765	⋆59	0.0000%	13	Optimal	2765	2765	119	0.00%	Optimal
5	170	9	4766	4766	⋆25	0.0000%	7	Optimal	4766	4766	54	0.00%	Optimal
6	194	9	3745	3745	475	0.0000%	105	Optimal	3745	3745	⋆428	0.00%	Optimal
7	291	9	5506	5506	⋆4629	0.0000%	72	Optimal	5506	5506	6069	0.00%	Optimal
8	97	13	6617	6617	⋆17	0.0000%	1	Optimal	6617	6617	65	0.00%	Optimal
9	120	13	9328	9328	⋆18	0.0000%	1	Optimal	9328	9328	231	0.00%	Optimal
10	125	13	8799	8799	⋆21	0.0000%	1	Optimal	8799	8799	83	0.00%	Optimal
11	154	13	11929	11929	⋆39	0.0000%	1	Optimal	11929	11929	598	0.00%	Optimal
12	170	13	12719	12719	⋆108	0.0000%	1	Optimal	12719	12719	505	0.00%	Optimal
13	194	13	14806	14806	⋆83	0.0000%	1	Optimal	14806	14806	217	0.00%	Optimal
14	291	13	21246	21246	⋆2174	0.0000%	131	Optimal	21246	21840	7200	2.71%	Time
15	97	18	11844	11844	⋆10	0.0000%	1	Optimal	11844	11844	1775	0.00%	Optimal
16	120	18	16376	16376	⋆21	0.0000%	1	Optimal	16376	16642	7200	1.62%	Time
17	125	18	17812	17812	⋆40	0.0000%	1	Optimal	17812	17812	280	0.00%	Optimal
18	154	18	20691	20692	7200	⋆0.0048%	56	Time	20691	20726	7200	0.16%	Time
19	170	18	14888	14892	7200	⋆0.0268%	4	Time	14889	14903	7200	0.09%	Time
20	194	18	12513	12521	7200	⋆0.0600%	144	Time	12518	12527	7200	0.07%	Time
21	291	18	34957	34962	7200	⋆0.0286%	80	Time	34961	35519	7200	1.59%	Time
22	97	20	12024	12024	⋆58	0.0000%	1	Optimal	12024	13158	7200	9.43%	Time
23	120	20	18239	18239	⋆86	0.0000%	1	Optimal	18239	18629	7200	2.13%	Time
24	125	20	17100	17100	⋆99	0.0000%	1	Optimal	17100	17876	7200	4.53%	Time
25	154	20	22375	22375	⋆807	0.0000%	1	Optimal	22375	23850	7200	6.59%	Time
26	170	20	21868	21868	⋆663	0.0000%	1	Optimal	21868	23245	7200	6.29%	Time
27	194	20	23398	23398	⋆770	0.0000%	1	Optimal	23398	25146	7200	7.47%	Time
28	291	20	36922	36926	7200	⋆0.0108%	126	Time	36917	37866	7200	2.57%	Time
29	97	22	11788	11788	⋆79	0.0000%	1	Optimal	11788	12134	7200	2.93%	Time
30	120	22	19620	19621	7200	⋆0.0050%	133	Time	19621	20368	7200	3.80%	Time
31	125	22	20853	20853	⋆116	0.0000%	1	Optimal	20853	22172	7200	6.32%	Time
32	154	22	26729	26729	⋆828	0.0000%	1	Optimal	26729	27752	7200	3.82%	Time
33	170	22	25476	25476	⋆1026	0.0000%	28	Optimal	25476	26508	7200	4.05%	Time
34	194	22	31251	31252	7200	⋆0.0031%	154	Time	31251	31954	7200	2.24%	Time
35	291	22	39788	39790	7200	⋆0.0050%	90	Time	39788	41906	7200	5.32%	Time
36	97	24	18987	18990	7200	⋆0.0157%	117	Time	18990	19785	7200	4.19%	Time
37	120	24	21231	21231	⋆1014	0.0000%	1	Optimal	21231	21950	7200	0.00%	Time
38	125	24	22376	22376	⋆561	0.0000%	48	Optimal	22376	23495	7200	5.00%	Time
39	154	24	29058	29058	⋆326	0.0000%	1	Optimal	29058	31208	7200	7.39%	Time
40	170	24	31353	31353	⋆610	0.0000%	15	Optimal	31353	32733	7200	4.40%	Time
41	194	24	31961	31962	7200	⋆0.0031%	178	Time	31961	32422	7200	1.44%	Time
42	291	24	31907	31909	7200	⋆0.0062%	70	Time	31907	31943	7200	0.11%	Time

Source: **The author**.

In eleven instances, optimality was not proved by either solving method, although the B&P obtained the smallest gaps. It is interesting to show the bounds evolution of both solvers within the time limit. For all instances where B&P branched (Node > 1), the same behavior as described in the previous subsection was observed. This behavior is illustrated for instance #4 in Figure 21, which shows the evolution of the optimality gap according to Gurobi and the B&P algorithm. At the first node, the B&P upper bound reaches the optimal value and the lower bound gets very close to $Z_{optimal}$ (GAP = 0.38%), steadily increasing as the search tree deepens. In comparison, Gurobi starts with a large gap (5.41%) which takes much longer to decrease, not

being able to prove optimality when the CPU time limit is reached.

Figure 21 – Instance #14 gap evolution for Gurobi and B&P.



Source: **The author**.

5.3.4 Illustrative examples

Firstly, two scenarios and two cases for each scenario are explored for the 3U size nanosatellite in order to evaluate the consistency of the results obtained by the B&P method. Then, to show how the algorithm performs in a more complex example, is presented the mission planning for a 6U mission.

5.3.4.1 3U Mission

The scheduling parameters used as input for all scenarios and cases of the 3U mission were randomly generated as presented in Table 25, amounting to nine tasks that are labeled from 1 to 9. The remaining parameters are presented in Table 26.

- **Scenario A**: An orbit with half-maximum eclipse and attitude Nadir.
- **Scenario B**: An orbit with half-maximum eclipse and attitude RAM.
- **Case 1**: The battery is allowed to be fully drained, down to the ρ limit, with a maximum current rate of 1.2 A ($\gamma = 1.2$).
- **Case 2**: The battery SoC has a variation limit within the optimization period, and constraints (35h) and (35i) are enforced with a maximum current rate of 5 A ($\gamma = 5$).
- **Case 3**: The battery SoC is limited with constraints (35h) and (35i), and scenarios A and B are solved several times with different γ values, to asses its impact on mission value extraction.

Table 25 – Nanosatellite tasks data (3U mission).

j	1	2	3	4	5	6	7	8	9
u_{sj}	7	4	8	6	9	2	1	5	3
q_{sj}	0.96	1.07	0.8	1.43	0.95	0.98	0.77	1.08	1.6
$y_j^{\min}$	5	3	5	3	2	4	3	5	4
$y_j^{\max}$	6	10	12	5	13	5	5	8	6
$t_j^{\min}$	9	15	11	18	20	3	8	1	6
$t_j^{\max}$	40	32	39	18	31	24	19	32	9
$p_j^{\min}$	43	26	22	31	33	16	43	38	22
$p_j^{\max}$	79	182	38	124	62	173	80	83	168
$w_j^{\min}$	0	0	0	0	0	0	0	0	12
$w_j^{\max}$	195	195	190	195	195	195	195	195	195

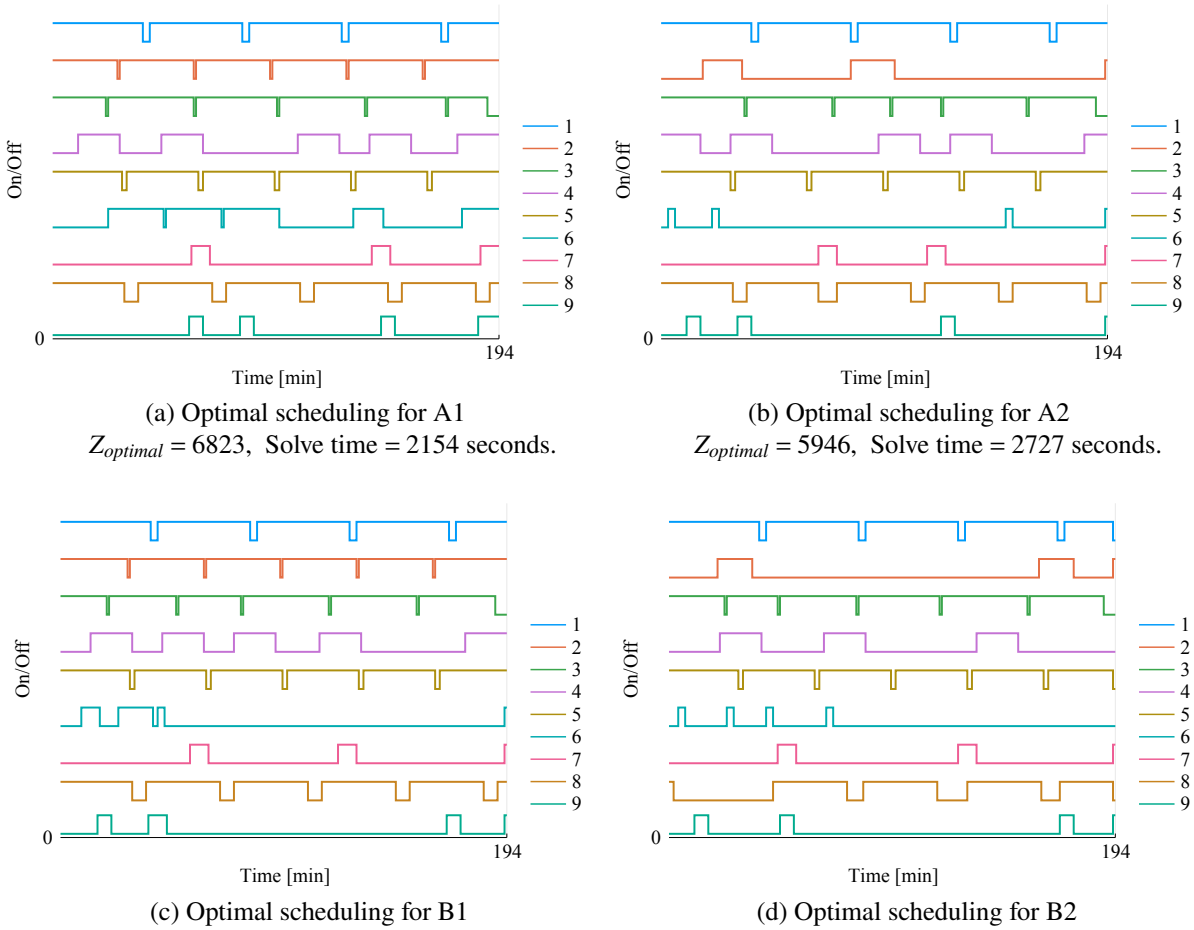
Source: **The author**.

Table 26 – Orbit and battery parameters (3U mission).

J	T	Q	V_b	e	SoC_0
9	194	5.6 Ah	3.6 V	0.9	75%

Source: **The author**.

Figure 22 – Branch-and-price scheduling results for all 3U scenarios.



(a) Optimal scheduling for A1
$Z_{optimal}$ = 6823, Solve time = 2154 seconds.

(b) Optimal scheduling for A2
$Z_{optimal}$ = 5946, Solve time = 2727 seconds.

(c) Optimal scheduling for B1
$Z_{optimal}$ = 6651, Solve time = 371 seconds.

(d) Optimal scheduling for B2
$Z_{optimal}$ = 5542, Solve time = 2562 seconds.

Source: **The author**.

Observe that nine tasks of high energy footprint are considered to be scheduled in this hypothetical 3U nanosatellite, and a battery with 5.6 Ah. Furthermore, the battery state of charge is not allowed to fall below 50% ($\rho = 0.5$) and the maximum discharge current rate is controlled. This is a realistic case since critical applications usually have a margin of safety and do not allow the battery to be fully discharged. Also, limiting the battery access current will improve its lifetime.

Figure 22 presents the optimal schedule obtained for each scenario. Notice that tasks are labeled from 1 to 9, the x-axis spans from time 0 to time T, and the y-axis indicates if the task is on or off for each unit of time. We have tasks 5, 3, and 1 with the highest priorities: 9, 8, and 7, respectively. The execution time of these tasks was maximized, being turned off only to comply with constraints on the maximum execution and period ranges. Notice that task 3 has an execution window parameter, $w_3^{\max} = 190$, hence it finishes execution before $T = 194$, in all 3U scenarios. Let us analyze task 1, for example, whose activation count must be within 5 and 6 times. It is easy to verify that these requirements were fulfilled, since task 1 initiated execution five times in each one of the scenarios.

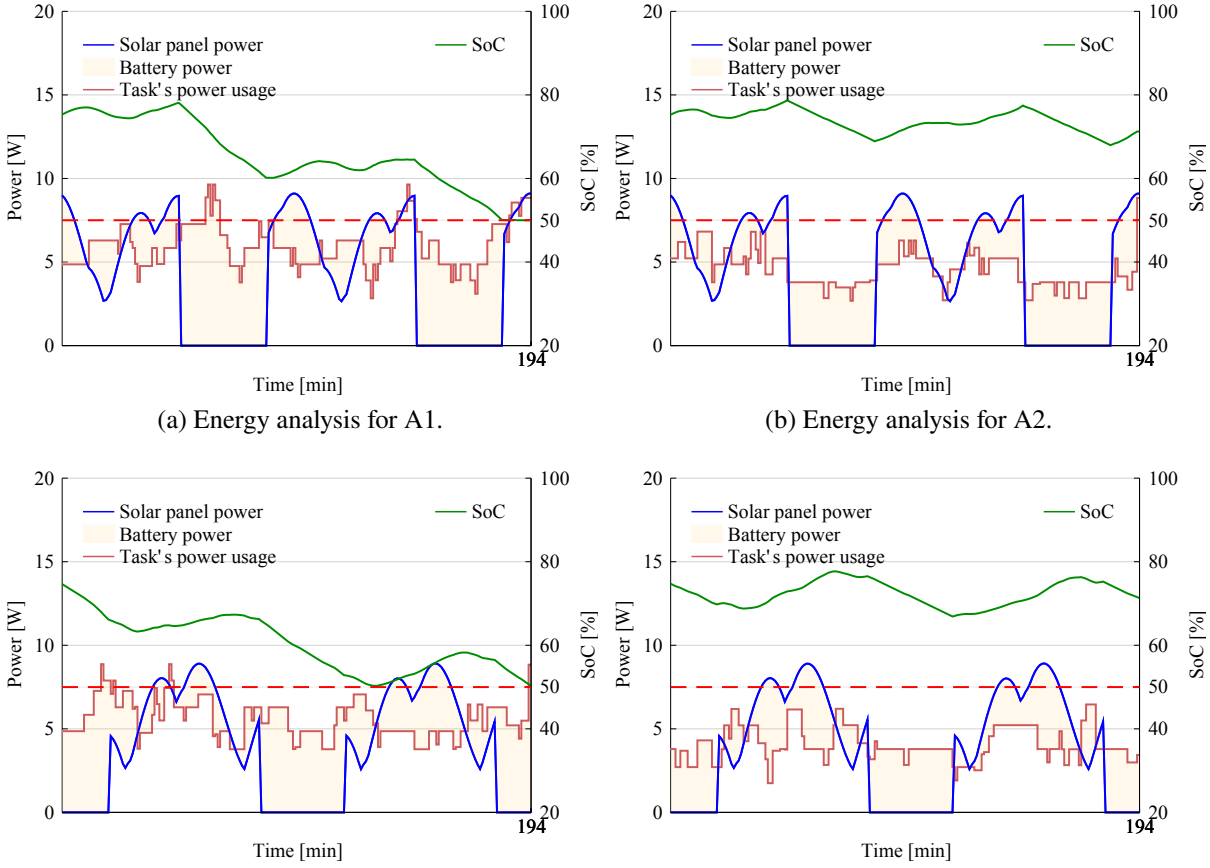
Furthermore, one can notice that tasks with lower priorities, such as tasks 2 and 6, are executed less frequently in cases A2 and B2, where energy is more scarce – the battery is not allowed to discharge more than 5% – than in cases A1 and B1. In fact, the final objective values of A2 and B2 are considerably smaller than for A1 and B1, respectively 12.89% and 16.67%. These are precisely the results expected from the scheduling formulation. It can also be observed that scenario B, with a RAM attitude, results in smaller objective values when compared within cases. This sort of analysis can be crucial for the engineer in the satellite designing phase, where the attitude is yet to be determined.

Figure 23 presents the results regarding energy management for all 3U cases. In blue, the calculated power input coming directly from the solar panels is shown, considering that these results refer to two full orbits and, therefore, two eclipse periods with zero power input. Moreover, it becomes clear how distinguishably different the input energy wave-forms are between cases A and B, with distinct attitudes. The battery power shown in these results can be either coming from or going to the battery, depending whether the task's power consumption rises above or falls below the energy input in a given minute. One can immediately observe a correlation between this power flow and the SoC since, when power consumption surpasses power input, the SoC falls and vice versa. It is particularly interesting to notice how the SoC steadily decreases during eclipse time.

As expected, in cases A1 and B1, the battery was fully drained down to the ρ limit of 50%, although never going below it. However, for cases A2 and B2, where constraints (35h) and (35i) are enforced, the battery acts more as a balancing factor between periods of energy input and eclipse time. This characterizes a much more farsighted and desirable behavior regarding battery access. In fact, the decision-maker could easily constrain, even more, the allowed variation, say to 1%, and always maintain the battery level above 80 or 90% to sustain an even higher margin

of safety. Figure 24 presents a histogram of the SoC to further analyze its dynanic.

Figure 23 – Branch-and-price power results for all 3U scenarios.



(a) Energy analysis for A1.

(b) Energy analysis for A2.

(c) Energy analysis for B1.

(d) Energy analysis for B2.

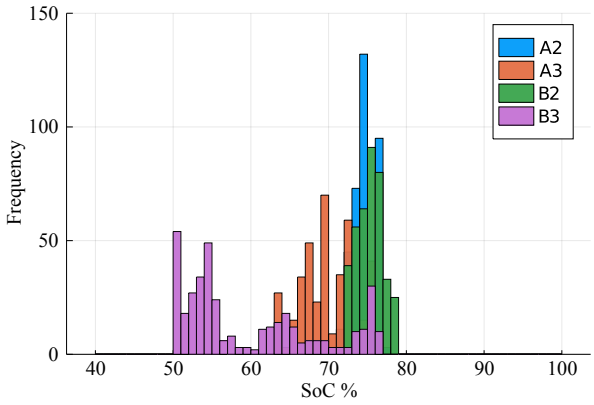
Source: **The author**.

Finally, for case 3, Figure 25 shows the impact of six distinct γ limits on the objective value. The result reveals that an aggressive limitation on the battery access (γ between 0.3 and 2.0) will result in a trade-off with mission value extraction, by means of which the decision-maker can balance battery lifetime and short-term gains.

5.3.4.2 6U Mission

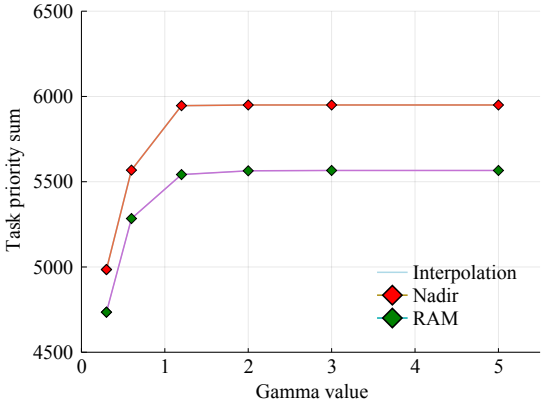
In this example, 18 tasks with high energy impact (labeled 1 to 18) are considered such as to represent a 6U mission with three consecutive orbits ($T = 291$). Table 27 and Table 28 present the scheduling parameters which were generated at random.

Figure 24 – SoC histogram for 3U scenario .



Source: **The author**.

Figure 25 – γ variation impact on the objective value – see expression (35a).



Source: **The author**.

Table 27 – Nanosatellite task data (6U mission).

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
u_{sj}	12	10	6	15	16	18	13	17	5	7	3	2	1	11	4	8	9	14
q_{sj}	3.05	1.8	1.6	0.65	1.94	3.42	2.22	2.16	2.04	2.55	3.06	1.51	1.71	1.61	2.89	0.59	1.3	1.24
$y_j^{\min}$	5	5	5	2	3	5	2	5	2	4	6	1	5	5	3	7	1	2
$y_j^{\max}$	14	20	16	11	8	18	2	19	17	16	15	7	16	11	9	10	4	2
$t_j^{\min}$	4	11	10	21	3	16	30	24	19	22	19	3	15	15	3	18	7	16
$t_j^{\max}$	44	15	48	43	60	40	35	32	57	56	23	70	25	65	63	22	26	29
$p_j^{\min}$	17	35	24	55	66	39	67	35	64	65	39	64	70	16	26	34	12	39
$p_j^{\max}$	280	132	150	56	188	260	105	239	265	123	51	127	258	211	105	254	120	260
$w_j^{\min}$	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0
$w_j^{\max}$	292	292	292	292	292	292	292	233	292	292	292	292	292	292	292	292	292	292

Source: **The author**.

A near-optimal schedule with optimality gap equal to 0.42% was obtained for this example after 7200 seconds of computation time, which is presented in two parts: part one in Figure 26a

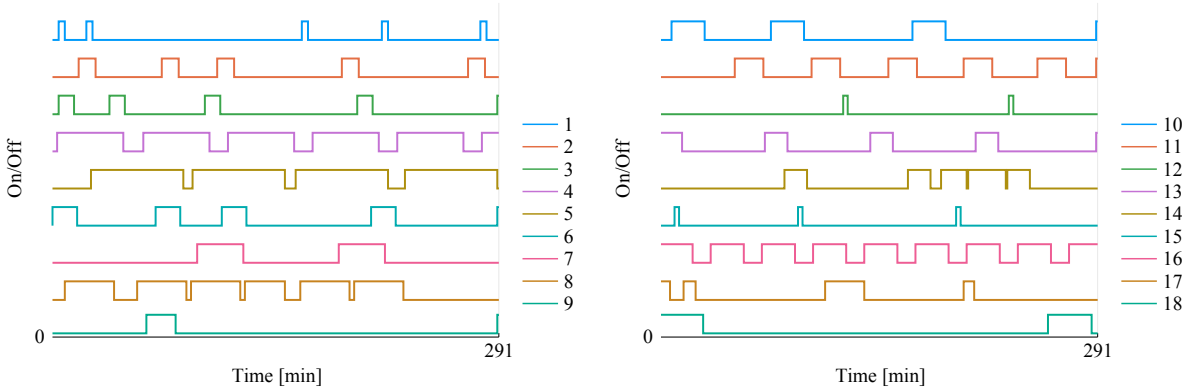
Table 28 – Scheduling parameters (6U mission).

J	T	Q	V_b	e	SoC_0	ρ	γ
18	291	10 Ah	3.6 V	0.9	85%	0.5	2 A

Source: **The author**.

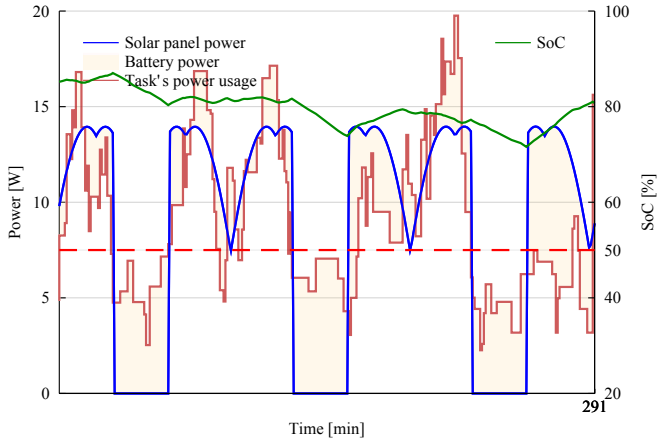
shows the schedule for tasks 1 to 9; and part two in Figure 26b depicts the schedule for tasks 10 to 18. Tasks 5 and 8, for instance, have windows of execution that are strictly followed by the scheduler. Task 18, despite having one of the highest priorities, is executed only two times as it should, given that its maximum number of startups is set at 2. Therefore, we can again verify the consistency of the scheduling results. The energy analysis, in Figure 26c, presents a result similar to the previous cases, except that now we have a longer eclipse time and a greater power input – reaching more than 12W.

Figure 26 – Branch-and-price results for a 6U mission, obtained when it reaches the time limit of 7200 seconds with $gap = 0.42\%$.



(a) Tasks 1 to 9.

(b) Tasks 10 to 18.



(c) Energy analysis.

Source: **The author**.

5.4 DISCUSSION

Scheduling a nanosatellite mission is especially challenging, given the limited availability of resources, the increasing implementation of the standard, and the consequent demand for better extraction of mission value and guarantees for quality-of-service. In this chapter, the state-of-the-art was advanced by devising an effective solution strategy to cope with large and complex instances of the problem. A Dantzig-Wolfe decomposition was performed to develop a novel profile-based formulation, and resorted to a column generation procedure to generate feasible schedules on demand, for each task. Furthermore, to guarantee the integrality of the solutions using the schedules obtained by column generation, designing a problem-specific branching scheme and performed changes to the restricted master problem and subproblems, resulting in a branch-and-price algorithm.

The results of computational experiments with randomly generated, but realistic instances showed an improvement of 70% on overall solution time compared to the standard approach, which consists in solving a MILP problem with a commercial solver. In addition, several illustrative examples based on realistic missions were scheduled, with the results corroborating the consistency of the proposed branch-and-price approach. Moreover, a significant reduction in optimality gaps was observed in all instances when using the proposed approach, allowing the decision-maker to quickly obtain near-optimal solutions for best mission extraction of value.

6 CONCLUSION

The idea of estimating an offline task schedule can serve as a basis for a nanossatelite project. Based on its results, the engineer, for example, can choose to include or remove a payload, prohibit the execution of tasks simultaneously, estimate the occurrence of safe modes of operation, consider the time frame between the launch and full nominal operation, worst-case scenarios, and so on. The work presented here has addressed this scheduling problem, tackling its specific necessities and requisites.

The nanossatelite scheduling problem characterization presented allowed to clearly understand the complexities, limitations and requisites related to planning a successful nanosatellite mission. Complementary, the comprehensive literature review presented answered several questions regarding how the problem has or has not being addressed by the scientific community. Therefore, it is believed that the specific objective (1) has been fulfilled.

The base model IP formulation has fully addressed objective (2), establishing a scheduling formulation generic enough to handle different task resolutions. The task considered could be an execution made by one of the embedded systems (i.e., an operating system task) or even the operation of an entire payload. Therefore, the function performed by the task (i.e., data download, beacon) or by the satellite itself (i.e., earth observation, communications network) is indifferent to the optimization model created, allowing to readily schedule any tasks.

The extended MILP formulation has fully addressed objective (3), by modeling a battery and formulating its energy access as a fuzzy constraint, allowing the scheduling formulation to exceed in an optimal way (decided by the optimization solver) the energy limit previously imposed by the solar panels power input at any given moment. The benefit of considering the battery was shown in the increased value of the tasks priority sum when compared with the model without a battery. Furthermore, the battery allows for scheduling critical tasks in periods of eclipse time (without energy input from solar pannels).

The multy-orbit scheduling was also a importatn adittion, because although the cyclical regime appears in some cases, the multi-orbit formulation serves to deal with changes in the battery's energy level, that could cause periodic payload regimes to change over time to achieve the feasibility of the optimization problem. In addition to being closely related to the execution of tasks, the battery level is dependent on the input energy available in the satellite. In closed orbits, the variation in the input is not perceptible. However, when analyzing the satellite's useful life, there may be a significant variation in the energy input levels due to disturbances in orbit.

The Danzig-Wolfe decomposition and branching strategy allowed scaling to finer time granularity and greater amount of tasks, completing the objective (4). Finally, comparisons aimed at meeting objective (5) showed that the branch-and-price methodology is, on average, 88% faster in obtaining optimal results compared to the computational times of the MILP solved with a commercial solver. In conclusion, a robust and realistic framework for optimal offline scheduling of nanosatellite missions is believed to have been achieved, which empowers the

decision-maker to readily schedule any mission in one or multiple subsystems of a nanosatellite.

6.1 FURTHER RESEARCH DIRECTIONS

In circuits where the solar panels are directly coupled, the satellite's power demand should change following the solar irradiation levels reaching the photo-voltaic panels (SARAVANAN; BABU, 2016; SLONGO et al., 2018), resulting in a maximum energy collection, also known as Maximum Power Point (MPP). This load behavior maintains the MPP of the solar panels and characterizes a state of maximum energy harvesting. Therefore, it is believed that a great contribution would be to create a computationally lightweight scheduling formulation enabling the Maximum Power Point Tracking (MPPT) in real time on-board the spacecraft. This formulation would be complementary to the methodology presented in this thesis, leveraging the offline schedule obtained here and encompassing more fine grained tasks – operating system tasks, for instance.

Another research path would be to explore hybrid methods with Machine Learning (ML) which could further improve the B&P solving time. Recent research has successfully explored ML models to predict: the final optimal value of decision variables (FURIAN et al., 2021), which variables select for branching (LODI; ZARPELLON, 2017), which columns to include in the re-optimization of the RMP and even to find heuristic-cuts (MORABIT; DESAULNIERS; LODI, 2021; PESCHIERA et al., 2021). Another promising idea is to train a ML model on task parameters and the relevant components of the optimal dual solution $(\bar{\mu}_j, \bar{\pi}_t, \bar{\nu}_t, \bar{\zeta}^k_{j,0}, \bar{\zeta}^k_{j,1} : \forall j \in \mathcal{J}, \forall t \in \mathcal{T})$, or features of them, to predict $\widetilde{c}_j$ and provide bounds for the pricing procedure, potentially decreasing solve time in this step.

REFERENCES

ALVAREZ, Aldair; MUNARI, Pedro. An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. **Computers & Operations Research**, v. 83, p. 1–12, 2017. ISSN 0305-0548. DOI: 10.1016/j.cor.2017.02.001.

BARNHART, Cynthia et al. Branch-and-price: Column generation for solving huge integer programs. **Operations research**, INFORMS, v. 46, n. 3, p. 316–329, 1998.

BEHRMANN, Gerd; LARSEN, Kim G.; RASMUSSEN, Jacob I. Priced Timed Automata: Algorithms and Applications. In: [s.l.: s.n.], 2005. P. 162–182. DOI: 10.1007/11561163_8. Available from: http://link.springer.com/10.1007/11561163%7B%5C_%7D8.

BELLMAN, Richard E; ZADEH, Lotfi Asker. Decision-making in a fuzzy environment. **Management Science**, INFORMS, v. 17, n. 4, b-141–b-164, 1970. DOI: 10.1287/mnsc.17.4.B141.

BIANCHESSI, Nicola et al. A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites. **European Journal of Operational Research**, Elsevier, v. 177, n. 2, p. 750–762, 2007. DOI: 10.1016/j.ejor.2005.12.026.

BISGAARD, Morten et al. Battery-aware scheduling in low orbit: the GomX–3 case. **Formal Aspects of Computing**, Springer, v. 31, n. 2, p. 261–285, 2019. DOI: 10.1007/s00165-018-0458-2.

BODINGTON, Charles E; BAKER, Thomas E. A history of mathematical programming in the petroleum industry. **Interfaces**, INFORMS, v. 20, n. 4, p. 117–127, 1990.

CAMPONOGARA, Eduardo et al. A continuous-time formulation for optimal task scheduling and quality-of-service assurance in nanosatellites. **Computers & Operations Research**, v. 147, p. 105945, 2022. ISSN 0305-0548. DOI: https://doi.org/10.1016/j.cor.2022.105945.

CASTRO, Monilito; STRAUB, Jeremy. Nanosatellite scheduling using a dictionary module and a ‘useful trick’with coded unsigned integers. In: IEEE. IEEE Aerospace Conference. [S.l.: s.n.], 2015. P. 1–7.

CASTRO, Pedro M. Tightening piecewise McCormick relaxations for bilinear problems. **Computers & Chemical Engineering**, Elsevier, v. 72, p. 300–311, 2015. DOI: 10.1016/j.compchemeng.2014.03.025.

CHEN, Xiaoyu et al. A mixed integer linear programming model for multi-satellite scheduling. **European Journal of Operational Research**, Elsevier, v. 275, n. 2, p. 694–707, 2019. DOI: 10.1016/j.ejor.2018.11.058.

CHIN, Alexander et al. CubeSat: the pico-satellite standard for research and education. In: AIAA Space 2008 Conference & Exposition. [S.l.: s.n.], 2008. DOI: 10.2514/6.2008-7734.

CHU, Xiaogeng; CHEN, Yuning; TAN, Yuejin. An anytime branch and bound algorithm for agile earth observation satellite onboard scheduling. **Advances in Space Research**, Elsevier, v. 60, n. 9, p. 2077–2090, 2017. DOI: 10.1016/j.asr.2017.07.026.

COOPER, I Diane. What is a "mapping study?" **Journal of the Medical Library Association: JMLA**, Medical Library Association, v. 104, n. 1, p. 76, 2016.

CRITCHLEY-MARROWS, Joshua; ISACSSON, Martin; GÅRDEBÄCK, Agnes. Applications of Linear Programming Techniques to Satellite Power Management and Scheduling. **Transactions of the Japan Society for Aeronautical and Space Sciences**, v. 17, n. 1, p. 57–61, 2019. DOI: 10.2322/tastj.17.57.

CUI, Jintian; ZHANG, Xin. Application of a Multi-Satellite Dynamic Mission Scheduling Model Based on Mission Priority in Emergency Response. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 19, n. 6, p. 1430, 2019. DOI: 10.3390/s19061430.

CURTIS, Howard D. **Orbital Mechanics for Engineering Students**. Third. Boston: Butterworth-Heinemann, 2014. P. 751. ISBN 978-0-08-097747-8. DOI: 10.1016/C2011-0-69685-1.

DANTZIG, George B.; WOLFE, Philip. Decomposition principle for linear programs. **Operations Research**, INFORMS, v. 8, n. 1, p. 101–111, 1960.

DENG, Boyu et al. Two-phase task scheduling in data relay satellite systems. **IEEE Transactions on Vehicular Technology**, IEEE, v. 67, n. 2, p. 1782–1793, 2017. DOI: 10.1109/TVT.2017.2763150.

DIAS, Lucas Pfeiffer Salomâo; BARBOSA, Jorge Luis Victoria; VIANNA, Henrique Damasceno. Gamification and serious games in depression care: A systematic mapping study. **Telematics and Informatics**, Elsevier, v. 35, n. 1, p. 213–224, 2018.

DUNNING, Iain; HUCHETTE, Joey; LUBIN, Miles. JuMP: A Modeling Language for Mathematical Optimization. **SIAM Review**, v. 59, n. 2, p. 295–320, 2017. DOI: 10.1137/15M1020575.

EDALAT, Neda; MOTANI, Mehul. Energy-aware task allocation for energy harvesting sensor networks. **EURASIP Journal on Wireless Communications and Networking**, SpringerOpen, v. 2016, n. 1, p. 28, 2016. DOI: 10.1186/s13638-015-0490-3.

FILHO, Edemar Morsch et al. Irradiation Flux Modelling for Thermal–Electrical Simulation of CubeSats: Orbit, Attitude and Radiation Integration. **Energies**, v. 13, n. 24, 2020. ISSN 1996-1073. DOI: 10.3390/en13246691.

FORREST, John et al. COIN-OR/CBC: Version 2.9.9. [S.l.], July 2018. DOI: 10.5281/zenodo.1317566.

FRAIRE, Juan et al. Battery-aware contact plan design for LEO satellite constellations: The Ulloriaq case study. In: IEEE. 2018 IEEE Global Communications Conference (GLOBECOM). [S.l.: s.n.], 2018. P. 1–7.

FRAIRE, Juan A et al. On the scalability of battery-aware contact plan design for LEO satellite constellations. **International Journal of Satellite Communications and Networking**, Wiley Online Library, p. 193–204, 2020. DOI: 10.1002/sat.1374.

FRISKE, Marcelo Wuttig; BURIOL, Luciana S.; CAMPONOGARA, Eduardo. A Branch-and-Price algorithm for a compressor scheduling problem. **Computers & Industrial Engineering**, v. 116, p. 72–81, 2018. ISSN 0360-8352. DOI: https://doi.org/10.1016/j.cie.2017.12.022.

FU, Alvin C; MODIANO, Eytan; TSITSIKLIS, John N. Optimal energy allocation and admission control for communications satellites. **IEEE/ACM Transactions on Networking**, IEEE, v. 11, n. 3, p. 488–500, 2003. DOI: 10.1109/TNET.2003.813041.

FURIAN, Nikolaus et al. A machine learning-based branch and price algorithm for a sampled vehicle routing problem. **OR Spectrum**, Springer, v. 43, n. 3, p. 693–732, 2021.

GAREY, Michael R.; JOHNSON, David S. **Computers and Intractability; A Guide to the Theory of NP-Completeness**. USA: W. H. Freeman & Co., 1990. ISBN 0716710455.

GILMORE, D.G.; DONABEDIAN, M. **Spacecraft Thermal Control Handbook: Fundamental technologies**. [S.l.]: Aerospace Press, 2002. (Spacecraft Thermal Control Handbook). ISBN 9781884989117. Available from: https://books.google.com.br/books?id=-GYGlwG8PkUC.

GOMSPACE. **P110 Solar Panel**. v. -. [S.l.: s.n.], 2020. P. -. https://gomspace.com/shop/subsystems/power/p110-solar-panel.aspx. Accessed: 2020-01-14.

GONDZIO, Jacek; GONZÁLEZ-BREVIS, Pablo; MUNARI, Pedro. New developments in the primal–dual column generation technique. **European Journal of Operational Research**, Elsevier, v. 224, n. 1, p. 41–51, 2013.

GRISWOLD, William. How to read an engineering research paper. **URL: http://www. cs. ucsd. edu/˜ wgg/CSE210/howtoread. html (: 30.09. 2010 .)**, 2009.

GROSSMANN, Ignacio E.; LEE, Sangbum. Generalized Convex Disjunctive Programming: Nonlinear Convex Hull Relaxation. **Computational Optimization and Applications**, v. 26, p. 83–100, 2003. DOI: 10.1023/A:1025154322278.

GUROBI OPTIMIZATION, Inc. **Gurobi Optimizer Reference Manual**. [S.l.: s.n.], 2016. Available from: http://www.gurobi.com.

HAIJIAO, Wang et al. Online scheduling of image satellites based on neural networks and deep reinforcement learning. **Chinese Journal of Aeronautics**, Elsevier, v. 32, n. 4, p. 1011–1019, 2019. DOI: 10.1016/j.cja.2018.12.018.

HAYTON, James. **PhD: an uncommon guide to research, writing & PhD life**. [S.l.]: James Hayton, 2015.

HE, Lei et al. Hierarchical scheduling for real-time agile satellite task scheduling in a dynamic environment. **Advances in Space Research**, Elsevier, v. 63, n. 2, p. 897–912, 2019. DOI: 10.1016/j.asr.2018.10.007.

HERMANNS, Holger; KRČÁL, Jan; NIES, Gilles. How is your satellite doing? battery kinetics with recharging and uncertainty. **Leibniz Transactions on Embedded Systems**, v. 4, n. 1, 2017.

HU, Xiaoxuan et al. A branch and price algorithm for EOS constellation imaging and downloading integrated scheduling problem. **Computers & Operations Research**, v. 104, p. 74–89, 2019. ISSN 0305-0548. DOI: https://doi.org/10.1016/j.cor.2018.12.007. Available from: https://www.sciencedirect.com/science/article/pii/S0305054818303113.

JIA, Xiaohua et al. Collaborative data downloading by using inter-satellite links in LEO satellite networks. **IEEE Transactions on Wireless Communications**, IEEE, v. 16, n. 3, p. 1523–1532, 2017. DOI: 10.1109/TWC.2017.2647805.

KESHAV, Srinivasan. How to read a paper. **ACM SIGCOMM Computer Communication Review**, ACM New York, NY, USA, v. 37, n. 3, p. 83–84, 2007.

KØRVELL, August; DEGN, Kristoffer. **Designing a Tool-Chain For Generating Battery-Aware Contact Plans Using UPPAAL**. Selma Lagerløfs Vej 300, 9220 Aalborg øst: [s.n.], 2019.

LAM, Jason T; RIVEST, François; BERGER, Jean. Deep Reinforcement Learning for Multi-satellite Collection Scheduling. In: SPRINGER. INTERNATIONAL Conference on Theory and Practice of Natural Computing. [S.l.: s.n.], 2019. P. 184–196. DOI: 10.1007/978-3-030-34500-6_13.

LI, Jianwei et al. Analysis of battery lifetime extension in a SMES-battery hybrid energy storage system using a novel battery lifetime model. **Energy**, Elsevier, v. 86, p. 175–185, 2015. DOI: 10.1016/j.energy.2015.03.132.

LODI, Andrea; ZARPELLON, Giulia. On learning and branching: a survey. **Top**, Springer, v. 25, n. 2, p. 207–236, 2017.

LÜBBECKE, Marco E; DESROSIERS, Jacques. Selected topics in column generation. **Operations Research**, INFORMS, v. 53, n. 6, p. 1007–1023, 2005.

MAENHOUT, Broos; VANHOUCKE, Mario. Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem. **Journal of Scheduling**, Springer, v. 13, n. 1, p. 77–93, 2010. DOI: 10.1007/s10951-009-0108-x.

MARCELINO, Gabriel Mariano et al. A Critical Embedded System Challenge. **IEEE Latin America Transactions**, v. 18, n. 2, p. 249–256, 2020. DOI: 10.1109/TLA.2020.9085277.

MARCHAND, Hugues et al. Cutting planes in integer and mixed integer programming. **Discrete Applied Mathematics**, Elsevier, v. 123, n. 1-3, p. 397–446, 2002. DOI: 10.1016/S0166-218X(01)00348-1.

MENG, Huan et al. Multi-Satellite Resource Scheduling Based on Deep Neural Network. In: INTERNATIONAL Joint Conference on Neural Networks (IJCNN). [S.l.: s.n.], 2019. P. 1–7. DOI: 10.1109/IJCNN.2019.8852044.

MORABIT, Mouad; DESAULNIERS, Guy; LODI, Andrea. Machine-learning–based column selection for column generation. **Transportation Science**, INFORMS, v. 55, n. 4, p. 815–831, 2021.

MOSER, Clemens et al. Real-time scheduling with regenerative energy. In: 18TH Euromicro Conference on Real-Time Systems (ECRTS'06). [S.l.: s.n.], 2006. DOI: 10.1109/ECRTS.2006.23.

MUNARI, Pedro; GONDZIO, Jacek. Using the primal-dual interior point algorithm within the branch-price-and-cut method. **Computers & Operations Research**, Elsevier, v. 40, n. 8, p. 2026–2036, 2013.

NIES, Gilles et al. Mastering operational limitations of LEO satellites–The GomX-3 approach. **Acta Astronautica**, Elsevier, v. 151, p. 726–735, 2018. DOI: j.actaastro.2018.04.040.

NING, Gang; HARAN, Bala; POPOV, Branko N. Capacity fade study of lithium-ion batteries cycled at high discharge rates. **Journal of Power Sources**, Elsevier, v. 117, n. 1-2, p. 160–169, 2003. DOI: 10.1016/S0378-7753(03)00029-6.

PANG, Chee Khiang et al. Nano-satellite swarm for SAR applications: Design and robust scheduling. **IEEE Transactions on Aerospace and Electronic Systems**, IEEE, v. 51, n. 2, p. 853–865, 2015. DOI: 10.1109/TAES.2014.140077.

PEIDRO, David et al. Fuzzy optimization for supply chain planning under supply, demand and process uncertainties. **Fuzzy Sets and Systems**, Elsevier, v. 160, n. 18, p. 2640–2657, 2009. DOI: 10.1016/j.fss.2009.02.021.

PESCHIERA, Franco et al. A novel solution approach with ML-based pseudo-cuts for the Flight and Maintenance Planning problem. **OR Spectrum**, Springer, v. 43, n. 3, p. 635–664, 2021.

PISHVAEE, Mir Saman; RAZMI, Jafar. Environmental supply chain network design using multi-objective fuzzy mathematical programming. **Applied Mathematical Modelling**, Elsevier, v. 36, n. 8, p. 3433–3446, 2012. DOI: 10.1016/j.apm.2011.10.007.

POGHOSYAN, Armen; GOLKAR, Alessandro. CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions. **Progress in Aerospace Sciences**, Elsevier, v. 88, p. 59–83, 2017. DOI: 10.1016/j.paerosci.2016.11.002.

POP, Valer et al. State-of-the-art of battery state-of-charge determination. **Measurement Science and Technology**, v. 16, n. 12, r93–r110, 2005. DOI: 10.1088/0957-0233/16/12/r01.

RIGO, Cezar A.; SEMAN, L. O.; CAMPONOGARA, E.; MORSCH FILHO, E., et al. Mission plan optimization strategy to improve nanosatellite energy utilization and tasks QoS capabilities. In: IV IAA Latin American Cubesat Workshop (IAA-LACW'2020). Paris: International Academy of Astronautics. [S.l.: s.n.], 2020.

RIGO, Cezar Antônio; SEMAN, Laio Oriel; CAMPONOGARA, Eduardo; FILHO, Edemar Morsch, et al. A nanosatellite task scheduling framework to improve mission value using fuzzy constraints. **Expert Systems with Applications**, v. 175, p. 114784, 2021b. ISSN 0957-4174. DOI: 10.1016/j.eswa.2021.114784.

RIGO, Cezar Antônio; SEMAN, Laio Oriel; CAMPONOGARA, Eduardo; MORSCH FILHO, Edemar; BEZERRA, Eduardo Augusto. Task scheduling for optimal power management and quality-of-service assurance in CubeSats. **Acta Astronautica**, v. 179, p. 550–560, 2021. ISSN 0094-5765. DOI: 10.1016/j.actaastro.2020.11.016.

RIGO, Cezar Antônio; SEMAN, Laio Oriel; CAMPONOGARA, Eduardo; MORSCH FILHO, Edemar; BEZERRA, Eduardo Augusto; MUNARI, Pedro. A branch-and-price algorithm for nanosatellite task scheduling to improve mission quality-of-service. **European Journal of Operational Research**, v. 303, n. 1, p. 168–183, 2022. ISSN 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2022.02.040.

SADEGHI, Mehdi; HOSSEINI, Hossein Mirshojaeian. Evaluation of fuzzy linear programming application in energy models. **International Journal of Energy Optimization and Engineering (IJEOE)**, IGI Global, v. 2, n. 1, p. 50–59, 2013. DOI: 10.4018/ijeoe.2013010104.

SADYKOV, Ruslan et al. Primal Heuristics for Branch and Price: The Assets of Diving Methods. **INFORMS Journal on Computing**, v. 31, n. 2, p. 251–267, 2019. DOI: 10.1287/ijoc.2018.0822.

SAFI, Mohammadreza; MALEKI, H.; ZAEIMAZAD, E. A note on the Zimmermann method for solving fuzzy linear programming problems. **Iranian Journal of Fuzzy Systems**, v. 4, p. 31–45, Oct. 2007. DOI: 10.22111/IJFS.2007.369.

SAMSUNG SDI CO., Ltd. **NLithium-ion rechargeable cell for power tools Model name: INR18650-25R**. [S.l.]: Energy Business Division, 2014.

SARAVANAN, S; BABU, N Ramesh. Maximum power point tracking algorithms for photovoltaic system–A review. **Renewable and Sustainable Energy Reviews**, Elsevier, v. 57, p. 192–204, 2016.

SEMAN, Laio Oriel; RIBEIRO, Brenda F., et al. An Energy-Aware Task Scheduling for Quality-of-Service Assurance in Constellations of Nanosatellites. **Sensors**, v. 22, n. 10, 2022. ISSN 1424-8220. DOI: 10.3390/s22103715. Available from: https://www.mdpi.com/1424-8220/22/10/3715.

SEMAN, Laio Oriel; RODRIGUES MACHADO, Victor Hugo, et al. A framework to estimate dwell time of BRT systems using fuzzy regression. **Journal of Intelligent & Fuzzy Systems**, v. 38, p. 5279–5293, 2020. DOI: 10.3233/JIFS-191904.

SHKOLNIK, Evgenya L. On the verge of an astronomy CubeSat revolution. **Nature Astronomy**, Nature Publishing Group, v. 2, n. 5, p. 374–378, 2018. DOI: 10.1038/s41550-018-0438-8.

SLONGO, L. K. et al. Energy-driven scheduling algorithm for nanosatellite energy harvesting maximization. **Acta Astronautica**, v. 147, p. 141–151, 2018. ISSN 00945765. DOI: 10.1016/j.actaastro.2018.03.052.

SONG, Yanjie et al. An emergency task autonomous planning method of agile imaging satellite. **EURASIP Journal on Image and Video Processing**, Springer, v. 2018, n. 1, 2018. DOI: 10.1186/s13640-018-0268-8.

TANAKA, Hideo; OKUDA, Tetsuji; ASAI, Kiyoji. On Fuzzy-mathematical programming. **Transactions of the Society of Instrument and Control Engineers**, The Society of Instrument and Control Engineers, v. 9, n. 5, p. 607–613, 1973. DOI: 10.9746/sicetr1965.9.607.

VANDERBECK, François; WOLSEY, Laurence A. Reformulation and decomposition of integer programs. In: 50 Years of Integer Programming 1958-2008. [S.l.]: Springer, 2010. P. 431–502.

VANDERBEI, Robert J. **Linear Programming**. Boston, MA: Springer US, 2001. v. 37. (International Series in Operations Research & Management Science). ISBN 978-1-4757-5664-7. DOI: 10.1007/978-1-4757-5662-3.

VERDEGAY, J. L. Fuzzy Mathematical Programming. **Fuzzy Inf and Decis Processes**, p. 231–237, 1982. ISSN 0453-4654. DOI: 10.9746/sicetr1965.9.607.

VILLACORTA, Pablo J et al. FuzzyLP: an R package for solving fuzzy linear programming problems. In: GRANULAR, Soft and Fuzzy Approaches for Intelligent Systems. [S.l.]: Springer, 2017. P. 209–230.

WAN, Shu-Ping; QIN, Ying-Li; DONG, Jiu-Ying. A hesitant fuzzy mathematical programming method for hybrid multi-criteria group decision making with hesitant fuzzy truth degrees. **Knowledge-Based Systems**, Elsevier, v. 138, p. 232–248, 2017. DOI: 10.1016/j.knosys.2017.10.002.

WAN, Shu-Ping; WANG, Feng, et al. An intuitionistic fuzzy linear programming method for logistics outsourcing provider selection. **Knowledge-Based Systems**, Elsevier, v. 82, p. 80–94, 2015. DOI: 10.1016/j.knosys.2015.02.027.

WANG, Jianjiang; DEMEULEMEESTER, Erik; HU, Xuejun; QIU, Dishan, et al. Exact and Heuristic Scheduling Algorithms for Multiple Earth Observation Satellites Under Uncertainties of Clouds. **IEEE Systems Journal**, IEEE, v. 13, n. 3, p. 3556–3567, 2018. DOI: 10.1109/JSYST.2018.2874223.

WANG, Jianjiang; DEMEULEMEESTER, Erik; HU, Xuejun; WU, Guohua. Expectation and SAA Models and Algorithms for Scheduling of Multiple Earth Observation Satellites Under the Impact of Clouds. **IEEE Systems Journal**, v. 14, n. 4, p. 5451–5462, 2020. DOI: 10.1109/JSYST.2019.2961236.

WANG, Jianjiang; ZHU, Xiaomin, et al. Dynamic scheduling for emergency tasks on distributed imaging satellites with task merging. **IEEE Transactions on Parallel and Distributed Systems**, IEEE, v. 25, n. 9, p. 2275–2285, 2013. DOI: 10.1109/TPDS.2013.156.

WANG, Yu; SHENG, Min, et al. Multi-resource coordinate scheduling for earth observation in space information networks. **IEEE Journal on Selected Areas in Communications**, IEEE, v. 36, n. 2, p. 268–279, 2018. DOI: 10.1109/JSAC.2018.2804045.

WERNERS, Brigitte. An interactive fuzzy programming system. **Fuzzy Sets and Systems**, Elsevier, v. 23, n. 1, p. 131–147, 1987. DOI: 10.1016/0165-0114(87)90105-9.

WU, Guohua; MA, Manhao, et al. Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks. **Journal of Systems Engineering and Electronics**, BIAI, v. 23, n. 5, p. 723–733, 2012. DOI: 10.1109/JSEE.2012.00089.

WU, Guohua; WANG, Huilin, et al. An adaptive Simulated Annealing-based satellite observation scheduling method combined with a dynamic task clustering strategy. **arXiv preprint arXiv:1401.6098**, 2014.

WU, Jian; ZHANG, Jiawei, et al. Research on Task Priority Model and Algorithm for Satellite Scheduling Problem. **IEEE Access**, IEEE, v. 7, p. 103031–103046, 2019. DOI: 10.1109/ACCESS.2019.2928992.

ZHAI, Xuejun et al. Robust satellite scheduling approach for dynamic emergency tasks. **Mathematical Problems in Engineering**, Hindawi, v. 2015, 2015. DOI: 10.1155/2015/482923.

ZHAO, Weihu et al. Immune and genetic hybrid optimization algorithm for data relay satellite with microwave and laser links. In: PROCEEDINGS of the Genetic and Evolutionary Computation Conference Companion. [S.l.: s.n.], 2019. P. 2008–2015. DOI: 10.1145/3319619.3326874.

ZHOU, Di et al. Collaborative Data Scheduling With Joint Forward and Backward Induction in Small Satellite Networks. **IEEE Transactions on Communications**, IEEE, v. 67, n. 5, p. 3443–3456, 2019. DOI: 10.1109/TCOMM.2019.2900316.

ZHU, Xiaomin et al. Fault-tolerant scheduling for real-time tasks on multiple earth-observation satellites. **IEEE Transactions on Parallel and Distributed Systems**, IEEE, v. 26, n. 11, p. 3012–3026, 2014. DOI: 10.1109/TPDS.2014.2363657.

ZIMMERMANN, H.-J. Fuzzy programming and linear programming with several objective functions. **Fuzzy Sets and Systems**, Elsevier, v. 1, n. 1, p. 45–55, 1978. DOI: 10.1016/0165-0114(78)90031-3.