

On-board Automated Mission Planning for Spacecraft Autonomy: A Survey

F. Cividanes, M. Ferreira, and F. Kucinskis

Abstract—Traditionally, space mission operations are based on preplanned activities fulfilled by the ground segment. Due to restricted visibility windows and communication delays, some space systems cannot be completely controlled from the ground in real-time. A way to overcome this issue is to equip the satellite flight software with an autonomous decision-making system such as onboard automated planning techniques. This paper presents a comprehensive survey on goal-based mission planning to increase the level of spacecraft autonomy. The most used onboard planning techniques for space missions such as temporal and hierarchical planning are presented in detail. We describe several onboard planners and compare them by design choices and performance results. Research trends in goal-based planning for the space domain are also outlined. This paper highlights the automated mission planning, whose state of the art has not been deeply explored by other surveys on autonomous spacecraft.

Index Terms—Automated mission planning, Onboard planning, Spacecraft autonomy, Space mission operations, Temporal and hierarchical planning.

I. INTRODUÇÃO

TRADICIONALMENTE, as missões espaciais são controladas pelo paradigma de operação por sequência de comandos [1]. O plano de operação de voo é elaborado em solo, o qual é transmitido periodicamente ao segmento espacial por intermédio de uma sequência de comandos que serão executados, em momentos predeterminados, pelo computador de bordo [2]. A telemetria de serviço é monitorada pela equipe de solo, que averigua a saúde da plataforma e as condições da segurança em bordo. Os planos de operação são gerados segundo os objetivos da missão e condicionados às restrições da engenharia de operação.

Embora constitua o paradigma vigente, esta forma de operação está presa à ideia de que todas as atividades do veículo espacial devem ser previstas pelo segmento solo. Isso traz limitações, como a impossibilidade de rever os planos a partir de mudanças inesperadas na situação dos equipamentos e a incapacidade de responder a eventos não previstos e/ou transitórios. Para lidar com essas limitações, uma das tendências na área espacial é buscar o aumento do nível de

autonomia operacional a bordo dos veículos espaciais [3].

Para tanto, o segmento espacial deve possuir alguma capacidade de tomada de decisões, permitindo que a equipe em solo consiga controlar o satélite a partir de objetivos a atingir, ao invés de sequências de comandos. O software de voo deixa de ser um mero executor de comandos e passa a ter inteligência computacional em bordo, não dependendo apenas da intervenção do segmento solo para direcionar suas ações.

Uma plataforma espacial autônoma deve ser capaz, até certo nível, de observar novos objetivos da missão, planejar suas atividades conforme as metas impostas, executar e monitorar as ações planejadas, detectar a presença de falhas e, nesses casos, implantar possíveis estratégias de recuperação [4]. Para isso, um novo software deve ser estruturado em bordo [5]. As tarefas relacionadas ao planejamento de operação são consideradas por diversos estudos ([4], [6], [7]) como fundamentais para alcançar a maior autonomia de veículos espaciais. O planejamento automatizado é a área de Inteligência Artificial (IA) que estuda a automação do processo de deliberação de ações por meio da computação [8].

É possível aplicá-la a missões espaciais com um planejador embarcado e uma linguagem de descrição do domínio, que tornam possível definir as atividades que levarão o sistema a cumprir as metas, considerando as restrições temporais e de recursos do satélite. Neste contexto, as instituições espaciais apostam na autonomia como forma de reduzir custos de operação e aumentar a capacidade de resposta do satélite.

Um trabalho pioneiro sobre planejamento embarcado é o *Remote Agent eXperiment* (RAX [9]), concebido para operar autonomamente a sonda *Deep Space One* (DS-1) a partir de objetivos recebidos de solo. Já o *Autonomous Sciencecraft Experiment* (ASE [10]) foi o primeiro software relacionado a uma missão de sensoriamento remoto a usar planejamento de missão em bordo (*Continuous Activity Scheduling, Planning, Execution and Replanning* - CASPER [6]), evidenciando que a autonomia não está restrita a missões de espaço profundo.

Embora o planejamento de missão seja considerado o cérebro de um satélite [11], os únicos dois trabalhos de revisão ([3], [12]) voltados à autonomia de veículos espaciais, não o escolheram como temática central da pesquisa. Uma vez que este tópico ainda não foi profundamente investigado, o presente artigo tem como objetivo estabelecer o estado da arte em técnicas de planejamento automatizado para controlar um satélite com autonomia operacional em bordo.

O artigo está organizado da seguinte maneira: na Seção II apresenta-se uma visão geral sobre autonomia de sistemas espaciais, na Seção III enfatiza-se o processo de transferência do planejamento em solo para o segmento espacial, na Seção IV são descritas as principais técnicas de planejamento em IA

F. S. Cividanes, Instituto Nacional de Pesquisas Espaciais - INPE, São José dos Campos, SP, Brasil, filipe.cividanes@inpe.br.

M. G. V. Ferreira, Instituto Nacional de Pesquisas Espaciais - INPE, São José dos Campos, SP, Brasil, mauricio.ferreira@inpe.br.

F. N. Kucinskis, Instituto Nacional de Pesquisas Espaciais - INPE, São José dos Campos, SP, Brasil, fabricio.kucinskis@inpe.br.

para aplicações embarcadas, na Seção V são apresentados os planejadores de bordo existentes, a Seção VI traz um estudo comparativo dos trabalhos, a Seção VII sumariza as tendências futuras de pesquisa e a Seção VIII traz as conclusões.

II. AUTONOMIA EM BORDO DE VEÍCULOS ESPACIAIS

As principais agências espaciais vêm pesquisando sobre autonomia em bordo através de planejamento embarcado há mais de duas décadas. Dentre os estudos pioneiros, cita-se o programa *New Millennium* [13] da *National Aeronautics and Space Administration* (NASA) e os programas *Aurora* [14] e *Project for On-Board Autonomy* (PROBA [15]) da *European Space Agency* (ESA). As missões que demandam por autonomia incluem sondas interplanetárias, satélites de sensoriamento remoto e missões científicas. Em geral, elas dispõem de um ou mais dos seguintes fatores: necessidade de menor tempo de reação, limitação da banda de comunicação, previsibilidade insuficiente quanto ao ambiente a ser explorado, latência na comunicação e pouco tempo de visibilidade com as estações em solo [16]. A tendência é que, quanto maior a complexidade de operação, mais decisivo seja o papel da autonomia para o sucesso da missão [2], [17].

Para determinação do nível de autonomia, consideram-se: tipo de missão, órbita, objetivos e prioridades, conceitos de operações e restrições de comunicação [18]. O grau a ser estabelecido depende das particularidades de cada missão, e não é algo do tipo 'tudo ou nada' [3]. Um estudo feito em [19], constatou que o nível de autonomia é alto para procedimentos de rotina, como as atividades de suporte à engenharia a bordo, mas ainda baixo para atividades de planejamento e escalonamento. Isso significa que quanto maior o nível de raciocínio necessário para a execução de uma tarefa, menor é o grau de autonomia adotado pelas missões.

A. Automação Versus Autonomia

Embora a literatura não defina rigorosamente o conceito de autonomia [20], é importante notar a distinção entre 'automação de operação' e 'autonomia do software de voo'. Basicamente, a automação visa reduzir esforço e custo de operação através da automatização no processo de geração de comandos em bordo. Ela provê instruções predeterminadas em solo como resposta a ações ou eventos ocorridos em órbita.

Isso vem sendo explorado há décadas com o uso de telecomandos ditos 'temporizados' (*time tagged telecommand*) e mais recentemente, através de comandos em resposta à ocorrência de eventos (*event action*) e comandos disparados a partir de uma posição orbital (*position tagged telecommand*) do satélite. Já em algumas missões da ESA, através de *On-board Control Procedures* (OBCP), dos quais se tratará adiante neste artigo. OBCPs são processos automatizados, mas não autônomos [21]. Sistemas automatizados não concebem escolhas próprias [22] e quando se depararam com situações não planejadas ficam inertes. Isso é adequado a missões de menor complexidade, que interagem com ambientes previsíveis [23]. Para missões que requerem maior autonomia do software do voo, entretanto, o satélite deve tomar decisões rápidas e complexas mesmo em situações não previstas.

Com a autonomia, um sistema espacial pode ser controlado, comandando-o para atingir um conjunto de objetivos. O próprio sistema transforma os objetivos em sequências de

ações que realizam cada meta [3]. A autonomia pode ser definida como a capacidade do satélite agir por conta própria dentro de certos limites [24]. A Tabela I sintetiza as características entre 'Autonomia' e 'Automação' no contexto de sistemas espaciais.

TABELA I
AUTONOMIA *VERSUS* AUTOMAÇÃO

Descrição	Autonomia	Automação
Paradigma	Orientada a objetivos, adaptativa e independente	Orientado a tarefas e inflexíveis. Utiliza procedimentos detalhados associados a eventos predefinidos
Ambiente	Opera em ambientes não totalmente previsíveis	Ambiente precisa ser bem caracterizado
Executor	O executor trabalha com comandos gerados a partir de um planejador embarcado	O executor trabalha com comandos residentes no software ou aqueles enviados por solo
Recursos	Gerenciamento dinâmico de recursos a bordo pelo planejador	Sem gerenciamento de recursos a bordo. Eles são alocados, antecipadamente, e com margens, pela equipe de solo

B. Riscos e Considerações Quanto à Adoção de Autonomia

O risco de operar sem a intervenção humana decorre não só da incerteza do uso de um novo sistema autônomo, mas também no valor elevado de uma missão [25]. Isso traz resistência por parte dos gerentes que veem a autonomia como possível risco à missão. Por outro lado, ela poderia adicionar confiabilidade por reduzir a complexidade da operação em solo [26], uma vez que a automação elimina a chance de falhas humanas [27]. Isso vem ocorrendo com os subsistemas de controle de atitude e de controle térmico que operam cada vez mais de forma autônoma. Para concepção de sistemas autônomos, entretanto, recomenda-se incrementar o nível de autonomia de forma gradual, amadurecendo o projeto à medida que se obtém herança de voo nessa tecnologia.

Práticas tradicionais de Verificação e Validação (V&V) não são apropriadas para software de voo com autonomia [28], [29]. Sistemas espaciais autônomos devem ser verificados e validados de forma muito completa, utilizando bancos de testes capazes de simular cenários detalhados [30]. Nessa linha de V&V que os trabalhos ([29], [31]-[34]) exploram suas pesquisas. Em [35], [36], sugerem-se a verificação por métodos formais para testes mais rigorosos. Apesar disso, o esforço adicional que leva à autonomia deve ser considerado investimento [26], pois custos podem ser reduzidos. Corroborar-se isso pelo satélite *Earth Observing One* (EO-1), em que se estimou a economia de um milhão de dólares por ano, a partir do início de sua operação autônoma [37].

C. Autonomia e suas Implicações ao Segmento Solo

Como mencionado em [38], devem ser consideradas todas as implicações para o segmento solo quando há aumento no nível de autonomia. Uma preocupação já expressada na literatura é a perda de competência dos operadores, decorrentes de longos períodos de operação autônoma de missões espaciais. Para evitar tal situação, recomenda-se treinar os operadores com simuladores, a fim de manter a competência não só nas operações rotineiras, mas

especialmente em situações de anormalidade. Um benefício direto da autonomia é que ela pode reduzir a quantidade de recursos humanos e financeiros, que são notadamente importantes para os países em desenvolvimento [7]. A redução de tarefas de solo possibilita que os operadores se dediquem às atividades de planejamento em longo prazo, ao invés de procedimentos de rotina do satélite [39], [40].

D. Níveis de Autonomia Operacional de uma Missão Espacial

O padrão [41] da *European Cooperation for Space Standardization* (ECSS) define quatro níveis de autonomia no que diz respeito à execução de operações nominais de uma missão espacial, como disposto na Tabela II. Eles indicam em ordem crescente o aumento da autonomia e a menor dependência do segmento solo.

TABELA II
CLASSIFICAÇÃO DE AUTONOMIA OPERACIONAL [41]

Nível	Descrição	Funções
E1	Execução de missão controlada pelo segmento solo, autonomia limitada para aspectos de segurança	Controle em tempo real para operações nominais. Execução de comandos temporizados para aspectos de segurança
E2	Execução de planos pré-planejados em solo	Capacidade de armazenar telecomandos temporizados com programação a bordo
E3	Execução em bordo de operação de missão de forma adaptativa	Operações autônomas baseadas em eventos. Execução de procedimentos de controle de operação em bordo (OBCP)
E4	Execução em bordo de operação orientada a objetivos	Operação baseada em objetivos com replanejamento em bordo

O nível E1 é utilizado em missões que têm bastante tempo de visibilidade e permite o controle em tempo quase real, como ocorre em um satélite geoestacionário que requer baixa autonomia [42]. Adota-se o conceito de *Flight Control Procedures* (FCP) que consiste no envio de seqüências de telecomandos e verificação da telemetria. O nível E2 é adotado na maioria das missões atuais, no qual se tem programação de comandos temporizados, a partir de uma seqüência de comandos predefinida em solo. Utiliza-se em missões cujo tempo de visibilidade é menor como os satélites de sensoriamento remoto. O nível E3 prevê procedimentos de controle de operação em bordo (OBCP). Na comunidade Europeia, somente satélites que dispõem de tecnologias avançadas e inseridos em missões científicas, ou de observação da Terra, o implementaram [43].

Os OBCPs tornam mais adaptativa a execução de procedimentos em bordo, sendo uma forma primitiva de operação em ciclo fechado [12]. São utilizados, entre outras situações, para a execução de 'seqüências de lançamento' (que comportam a abertura de painéis solares, aquisição inicial do Sol, o ato de ligar pela primeira vez os equipamentos da plataforma, etc.), comutar um equipamento de voo de nominal para redundante, alterar modos de operação da carga útil e executar ações de *Failure, Detection, Isolation and Recovery* (FDIR). Os procedimentos podem ser ativados diretamente por telecomandos ou por eventos ocorridos em órbita, como prevê a norma [44] da ECSS.

Para alcançar o nível E4, o satélite deve reagir a um evento não somente baseado em regras, mas utilizando outras capacidades presentes em bordo, como planejamento embarcado e motores de inferência de estados. Nesse nível, uma nova forma de operação espacial deve ser estabelecida. Visando maior entendimento desse paradigma emergente, apresenta-se a seguir os paradigmas existentes.

III. PARADIGMAS DE OPERAÇÃO DE MISSÕES ESPACIAIS

A. Operação por Sequência de Comandos

O paradigma atual de operação espacial é baseado na geração de seqüências de comandos previamente definidos pelos operadores da missão. A Fig. 1 traz uma representação conceitual deste paradigma [2]. Sua finalidade não é mapear fielmente os elementos contidos na arquitetura de operação de uma missão espacial, mas ser útil para um melhor entendimento. Nela, um 'Determinador de Objetivos', representado pelos usuários da missão, define os objetivos a serem atingidos pelo segmento espacial.

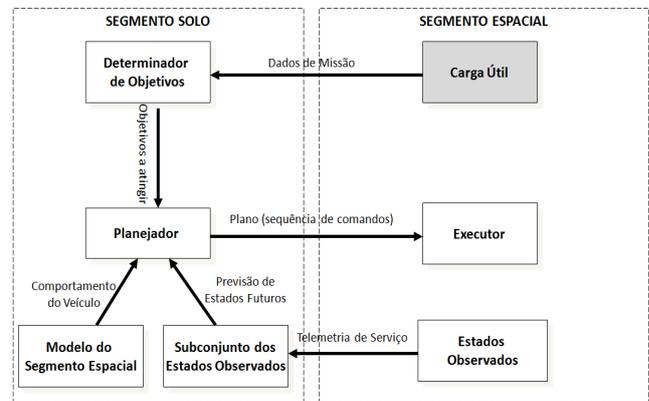


Fig. 1. Modelo conceitual da operação baseada em seqüência de comandos [2].

Cabe ao planejador transformar os objetivos em seqüências de comandos a serem transmitidas ao satélite. O planejador consulta o histórico de estados passados do veículo espacial, que lhe foram transmitidos via telemetria, e realiza, a partir de um modelo comportamental do satélite, uma inferência para determinar os estados futuros - aqueles que se espera encontrar em algum momento futuro de operação. Esse ciclo se repete durante a vida útil da missão a partir dos contatos subsequentes com a estação terrena.

Os níveis de E1 a E3 fazem uso dessa forma de operação baseada em comandos, que demandam o planejamento em solo, realizado com grande antecedência, e sem considerar possíveis mudanças no estado do satélite. Se novas oportunidades forem identificadas em órbita, não é possível adaptar o plano em tempo real devido às restrições temporais impostas pela comunicação solo-bordo. Isso caracteriza operação em ciclo aberto, com alta dependência do centro de controle e baixa autonomia do software de voo para tomada de decisões a bordo.

B. Operação Baseada em Objetivos

O paradigma de operação baseada em objetivos ([45]) está conceitualmente ilustrado pela Fig. 2. A ideia central é a

transferência do processo de planejamento, ou ao menos parte dele, para o segmento espacial. O satélite passa a receber objetivos a atingir, ao invés de comandos a executar. Fica a cargo dele a determinação de quais comandos levarão ao cumprimento dos objetivos. O planejador em conjunto com o software de voo tem acesso completo e, em tempo real, aos estados do satélite, o que potencialmente melhora a qualidade do processo de inferência de estados futuros. Além disso, os estados inferidos podem ser averiguados regularmente, disparando-se correções caso sejam detectados desvios entre o que foi previsto e o que foi aferido. Reduzem-se assim as limitações impostas pelo planejamento antecipado em solo e também pela falta de flexibilidade dos planos – uma vez que um plano pode ser adaptado a bordo, na medida do necessário. Tal forma de operação tem potencial de maximizar o uso dos recursos em bordo, aumentando o retorno científico da missão.

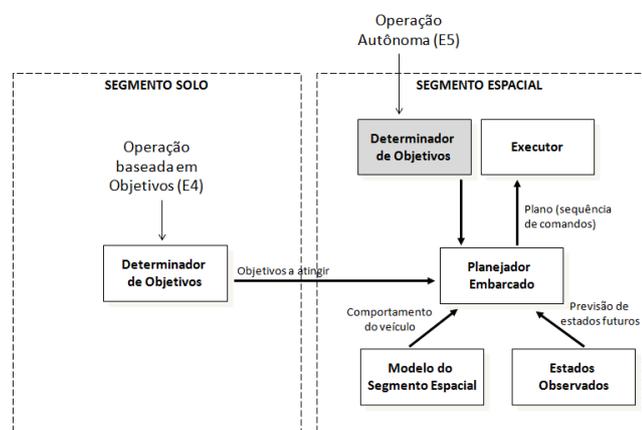


Fig. 2. Modelos conceituais da operação baseada em objetivos (E4) e operação autônoma do segmento espacial (E5) [2].

É possível, entretanto, evoluir além da operação baseada em objetivos. Como pode ser observado em cinza na Fig. 2, ao adicionar um determinador de objetivos ao segmento espacial, abre-se caminho para a operação autônoma do veículo espacial. Isso traz novas possibilidades à missão. Assumindo, por exemplo, que um algoritmo embarcado identifique a presença de queimadas nas imagens de um satélite de sensoriamento remoto, o segmento espacial pode requisitar outras observações sobre o alvo e encaminhá-las ao planejador embarcado na forma de novos objetivos. Isso aumenta substancialmente a capacidade de resposta do satélite e traz um nível de autonomia superior à classificação prevista pela ECSS. A Tabela III apresenta uma proposta de um novo nível de autonomia (E5), em que o segmento espacial é capaz de definir seus próprios objetivos. Há sua representação também na Fig. 2.

TABELA III
PROPOSTA DE UM NOVO NÍVEL DE AUTONOMIA

Nível	Descrição	Funções
E5	Operação autônoma do segmento espacial com autodeterminação dos objetivos em bordo	Possibilita que o planejador modifique autonomamente o plano, retirando ou adicionando objetivos a partir de eventos ou oportunidades de interesse da missão percebidos em órbita pelo determinador embarcado de objetivos

Para exemplos de determinador embarcado de objetivos, veja os trabalhos [40], [46], [47]. Eles são em geral algoritmos embarcados que visam o reconhecimento de padrões, como os usados para a classificação de imagens de satélites [48].

IV. ESTADO DA ARTE EM TÉCNICAS DE PLANEJAMENTO EM INTELIGÊNCIA ARTIFICIAL PARA APLICAÇÕES EMBARCADAS DA ÁREA ESPACIAL

Para alcançar um maior nível de autonomia do software de voo, faz-se necessário uma descrição do domínio espacial que modele as operações do satélite e um planejador embarcado que utilize alguma técnica de planejamento automatizado, descritos a seguir.

A. Definição Formal de um Problema de Planejamento Automático

O formalismo clássico para o problema de planejamento é definido pela tripla $P = \langle T, S_0, G \rangle$, onde [8]: T denota um sistema de transição de estados, S_0 é um estado inicial e G corresponde a um conjunto de estados objetivos. T , por sua vez, é representado pela seguinte tripla: $\langle S, A, \gamma \rangle$, onde: $S = \{s_0, s_1, \dots, s_n\}$ é o conjunto de estados; $A = \{a_0, a_1, \dots, a_m\}$ corresponde a um conjunto de ações; $\gamma = S \times A \rightarrow S$ denota a função de transição de estados. Uma solução para o plano P é composta por uma seqüência linear de ações que satisfaça os objetivos (G), tal que:

$$S_1 = \gamma(s_0, a_1), S_2 = \gamma(s_1, a_2), \dots, S_n = \gamma(s_{n-1}, a_n)$$

$$P = \{a_i, a_j, \dots, a_k\}, e S_n \in G$$

Tal seqüência de ações deve ser produzida pelo sistema de planejamento automatizado.

B. Abordagem Sequencial ou Integrada para Sistemas de Planejamento e Escalonamento (P&S)

Uma vez que se definam as ações do plano que satisfaçam aos objetivos (planejamento), é preciso de um escalonador para alocar tempo e recurso das atividades do plano. A maioria dos sistemas de planejamento mantém esses dois estágios de forma sequencial por simplificação: primeiro definem-se quais as ações, e depois as mesmas são posicionadas temporalmente e têm alocados a elas os recursos necessários para sua consecução. Porém, considerar as restrições impostas já na busca pelas ações, como nos sistemas P&S integrados, tende a melhorar o desempenho do sistema [49]. Nesse caso, a seqüência encontrada pelo planejador já considera o consumo de recursos e os momentos corretos de execução de cada comando. Na abordagem sequencial (não integrada), se um plano não for escalonável, o processo retorna novamente ao planejador para escolha de tarefas alternativas. Tal mecanismo pode se repetir ciclicamente, tornando o processo lento e ineficiente. Em aplicações reais, as etapas de P&S tendem a ser integradas [50].

C. Planejamento Centralizado ou Distribuído

Os sistemas são tipicamente integrados ao software de voo e adotam somente um planejador centralizado, como o *Planning and Scheduling* do RAX (RAX-PS). Outra solução é a adoção de planejamento distribuído, que promete melhorar o desempenho [4] e diminuir o espaço de busca, pois as

atividades podem ocorrer em paralelo. No entanto, isso traz maior complexidade para coordenação das tarefas de planejamento e para resolução de eventuais conflitos do plano.

Os sistemas que usaram a abordagem distribuída, também empregaram o conceito de multiagentes, vide os trabalhos: [4], [7]. Contudo, o planejamento distribuído é indicado em geral para agentes que estão em ambientes fisicamente distantes, como a operação de uma constelação de satélites autônomos - isto é tratado, por exemplo, em [25], [51], [52].

Para integrar o planejador embarcado à arquitetura do software de voo (vide [5], [28]), alguns trabalhos que usaram tanto a abordagem centralizada ([9], [10], [39], [53]), como a distribuída ([4]), organizaram sua arquitetura em três camadas hierárquicas. Tal estrutura em camadas é inspirada em trabalhos na área de robótica (*e.g.*, [54]) que estudam arquiteturas com princípios de controle autônomo.

D. Planejamento em lote versus planejamento contínuo

No planejamento realizado por lote (*batch*) - utilizado, por exemplo, em [9], [55] -, o tempo é dividido em vários horizontes de planejamento, cada qual com um período de tempo significativo. Quando o tempo se aproxima ao final do horizonte do plano corrente, o planejador projeta o estado futuro que o sistema estará no final da execução deste plano atual. A partir disso, formula-se um novo problema conforme os objetivos e os estados inferidos do próximo horizonte. Como o horizonte temporal é grande, o tempo de resposta do planejador é insatisfatório para atender novas demandas percebidas em órbita.

Em contrapartida, o planejamento contínuo, ou também conhecido como reparo iterativo ([56], [57]), permite o replanejamento em tempo real a partir de uma mudança no contexto do satélite. Ele visa minimizar o tempo de replanejamento de forma que, a qualquer momento, uma atualização incremental das metas, do estado atual ou do horizonte (com incrementos de tempo pequenos) pode mudar o estado do plano corrente. Isso deve prevalecer nos sistemas embarcados devido à menor latência na resposta. Além disso, mantendo um horizonte menor, a projeção dos estados futuros torna-se mais precisa, gerando planos de melhor qualidade.

E. Modelo de Planejamento para o Domínio Embarcado

Em um sistema de planejamento automático a inteligência está relacionada ao mecanismo de busca utilizado e ao conhecimento do domínio. Entende-se como domínio de um problema de planejamento o conhecimento necessário para sua solução. Ao modelar o domínio espacial, deve-se descrever o comportamento do satélite durante a sua operação. O modelo não especifica as ações necessárias para atingir os objetivos, mas define o efeito esperado de cada ação ou evento que pode ter no sistema modelado [12].

O modelo é consultado pelo planejador para ler o estado corrente da plataforma e, segundo os objetivos recebidos, criar um novo plano para execução. A partir de um mecanismo de inferência de operações, é possível o planejador exercitar variações no plano até encontrar uma que leve ao atendimento dos objetivos.

As variações incluem, por exemplo, remover, adicionar ou mover temporalmente as atividades do plano. Para que a inferência seja realizada a bordo, é preciso haver uma correlação entre os elementos do modelo e os elementos reais

do domínio embarcado. Os itens modelados devem estar relacionados a informações disponíveis ao software de voo por meio de uma interface.

Com relação à descrição do modelo, utiliza-se tipicamente uma linguagem para representação de estados, ações e objetivos e especificados por meio de uma representação declarativa. Todavia, linguagens do planejamento clássico, como *Planning Domain Description Language* (PDDL), não foram desenvolvidas para execução em ambientes embarcados. Uma vez que não existe ainda uma linguagem padronizada para área espacial [58], os projetistas têm concebido linguagens próprias para cada aplicação. Elas são em geral baseadas em intervalos de tempo, restrição de valores, quantidade de recursos consumidos, e outras restritas à modelagem por decomposição hierárquica de atividades.

Uma linguagem pioneira na área espacial é a *ASPEN Modeling Language* (AML) do planejador de operações executado em solo *Automated Scheduling and Planning Environment* (ASPEN [59]), cuja representação é voltada a ações, estados e restrições temporais para decomposição hierárquica de atividades. Outras são baseadas no conceito de 'linhas no tempo', ou do inglês *timelines* (intervalos, consumo de recursos no tempo e restrição de valores), tais como: a *New Domain Definition Language* (NDDL [60]) do planejador *Extensible Universal Remote Operations Planning Architecture* (EUROPA [60]), a *Internal State Inference Service modelling language* (ISIS-ml [61]) e a própria AML.

Note-se que os primeiros sistemas ([9], [10]) possuíam modelos distintos para as etapas de planejamento, de escalonamento e de execução do plano. Porém, conforme constatado por [4], [55], [62], diferenças na semântica dos modelos para representar um mesmo sistema levam a erros de projeto de difícil detecção, além de dificultar os testes. Isso abre a possibilidade para a ocorrência de falhas geradas por diferenças sutis entre os comportamentos previstos por cada modelo. Após a experiência negativa do RAX, alguns trabalhos sucessores [2], [4], [39], [63] tomaram o cuidado de manter um único modelo em sua arquitetura.

F. Formas de Representação dos Objetivos e Técnicas de Planejamento

Basicamente, a diferença entre um executor de um software de voo convencional (veja [64]) e um planejador embarcado é que o primeiro lida diretamente com comandos a partir do plano de operação realizado por solo, ao passo que o segundo trata os objetivos recebidos - sejam eles enviados pelo centro de controle ou gerados pelo próprio satélite - para encontrar a sequência de comandos que leva o satélite ao cumprimento destes objetivos.

Os planejadores da área espacial em geral lidam com os objetivos de duas formas. A primeira é traduzindo o objetivo recebido na forma de restrições de estados. A segunda é representá-lo como uma tarefa de alto nível, que seja decomposta em uma rede de tarefas hierarquicamente menores. Tais formas de representação são tratadas como 'Problemas de Satisfação de Restrições' (do inglês - *Constraint Satisfaction Problem* - CSP) e 'Rede de Tarefas Hierárquicas' (do inglês - *Hierarchical Task Network* - HTN), ambos os estudos na área de IA.

G. Planejadores Temporais Baseados em CSP

Muitos sistemas de planejamento embarcado da área espacial tratam os objetivos como um Problema de Satisfação de Restrições (CSP). Formalmente, um CSP é denotado por um conjunto finito de variáveis $X = \{x_1, \dots, x_n\}$; um conjunto finito de domínios de cada variável $\{D_1, \dots, D_n\}$, tal que $x_i \in D_i$; e um conjunto finito de restrições $\{C_1, \dots, C_m\}$. O objetivo final é atribuir valores para cada variável de forma que todas as restrições impostas sejam satisfeitas. Um CSP é dito consistente se tal solução existir. Há dois métodos principais de algoritmos que solucionam um CSP: busca construtiva e busca local (iterativos). De forma sumária, eles diferem em sua técnica de otimização: o primeiro tenta encontrar uma solução ótima, já o segundo tem seu foco na geração de planos rápidos [65]. Um ponto chave do planejador CSP, que difere do planejamento clássico, é a importância do mecanismo de inferência de estados. Em um domínio modelado na forma de CSP, os estados são definidos pelos valores de um conjunto de variáveis, e os objetivos consistem de um conjunto de restrições que tais valores devem obedecer.

Para que o CSP seja instanciado pelo planejador, é preciso que os objetivos sejam transformados em restrições sobre estados de variáveis do sistema a partir de um domínio no tempo. A ideia de representar um objetivo na forma de restrições do satélite pode ser vista com maior completude em [45]. A Fig. 3 ilustra o seguinte objetivo: 'o transmissor de dados deve ser ligado no momento 2500s e desligado em 3100s'.

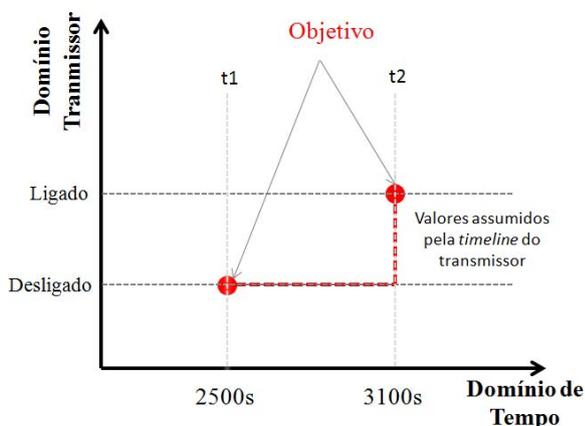


Fig. 3. Objetivo retratado como uma restrição binária de um estado (um recurso/timeline do sistema) no tempo (domínio de tempo).

A partir da Fig. 3, pode-se formular um CSP com as informações seguintes.

Dados os domínios:

$$D_{\text{transmissor}} = \{\text{ligado}, \text{desligado}\}$$

$$D_t = \{0, \dots\}$$

E as variáveis:

$$\text{transmissor} \in D_{\text{transmissor}}$$

$$\text{momento} \in D_t$$

A restrição que estabelece o objetivo é representada como:

$$R_{\text{transmissor-momento}} = (\text{ligado}, 2500; \text{desligado}, 3100).$$

Outros objetivos podem ser instanciados a partir de restrições que envolvam um ou mais estados. Por exemplo, a meta: 'ligar uma câmera no momento n ' em uma missão de

Observação da Terra envolve a alocação de diferentes recursos, tais como: potência, taxa de utilização da memória do gravador de dados, apontamento do satélite e consumo de combustível para eventuais manobras, etc.

A partir dessa forma de relacionamento entre estados e tempo é possível descrever um comportamento complexo através de conjuntos de restrições binárias que imponham ao CSP estados desejados para *timelines* e recursos em períodos específicos.

Um planejador temporal baseado em 'linhas do tempo' é definido como um conjunto de restrições sobre um conjunto de elementos temporais (pontos no tempo ou intervalos - os quais serão tratados adiante nesse artigo), atribuídos a algumas atividades. As soluções são representadas como *timelines* ou sequência de estados, definindo o comportamento desejado do sistema. Isso pode ser resolvido aplicando técnicas genéricas de CSP, embora algoritmos específicos para lidar com o gerenciamento de recursos possam ser utilizados [66].

Cada *timeline* representa a evolução de um estado de uma variável do sistema num dado horizonte, como o exemplo da Fig. 3. Sendo assim, qualquer atividade que altere recursos pode ser alocada numa *timeline*. A utilização de *timelines* em sistemas de planejamento e escalonamento demonstrou ser promissora em diversas aplicações reais, como, por exemplo, em sistemas espaciais autônomos [67].

Este conceito é particularmente aderente às aplicações espaciais, pois sua abstração está mais próxima da forma em que os problemas e as restrições são naturalmente representados na operação real de um satélite [58]. Sua característica permite aos planejadores temporais unificarem as etapas de planejamento e escalonamento [68]. Para informações adicionais sobre *timelines*, consulte [67], [69].

H. Planejadores Hierárquicos Baseados em HTN

Ao invés de buscar um estado-objetivo como no CSP, as técnicas de planejamento automatizado baseado em HTN tentam realizar um objetivo a partir da decomposição de tarefas de alto nível em subtarefas até chegarem às tarefas primitivas. Para cada tarefa não primitiva, o planejador escolhe um método aplicável e instancia-o para decompor a tarefa em subtarefas. Os métodos podem ter precondições para sua execução e efeitos de suas ações no sistema.

Além disso, pode haver restrições ou regras de como tarefas de mais baixo nível podem ou não se relacionar. Ao final, um plano válido é encontrado quando todas as restrições forem satisfeitas e restarem somente tarefas primitivas - aquelas que são executadas diretamente sem decomposição.

O problema de planejamento em HTN pode ser representado por 3-tuplas (s, T, D) , onde [49]: s representa os estados; T é uma rede de tarefas; $D = (A, M)$ denota o domínio do problema, onde: M é um conjunto de métodos e A é um conjunto de atividades (operadores de domínio) que indica como realizar as tarefas primitivas.

A técnica de HTN requer um conhecimento do domínio por especialistas para descrever a hierarquia das tarefas - funcionando, de certa forma, como heurísticas de busca -, mas sem depender de um mecanismo de busca refinado para encontrar as decomposições possíveis e ações associadas. Os métodos facilitam muito a solução do problema de planejamento, pois geram apenas planos que são soluções para

o problema. A HTN é adequada para domínios nos quais alguma representação hierárquica é desejável ou conhecida antecipadamente [70].

É útil para aplicações práticas devido à sua capacidade de diminuir o espaço de busca do problema. O planejamento em HTN é bastante eficiente em tempo de execução [49], o que é promissor para o ambiente embarcado. Sua aplicação em veículos espaciais autônomos permite diminuir drasticamente a complexidade do problema de planejamento [71]. A maioria dos sistemas que foram desenvolvidos para aplicações reais usaram a técnica de redução de tarefas [49], [50], porém ao menos na área espacial, ela vem perdendo espaço para os planejadores temporais.

Há uma prevalência dos planejadores hierárquicos em segregar o processo de planejamento e escalonamento em duas etapas por simplificação [49]. Na HTN, há uma carência para representação conjunta das restrições temporais e de recursos [50]. A Fig. 4 mostra um exemplo genérico de um objetivo expresso na forma de uma rede de tarefas (HTN) e a ilustração de um mecanismo de busca das atividades que satisfaçam este objetivo. A seta apontando para a direita representa a ordenação das subtarefas que nesse exemplo é de ordem total.

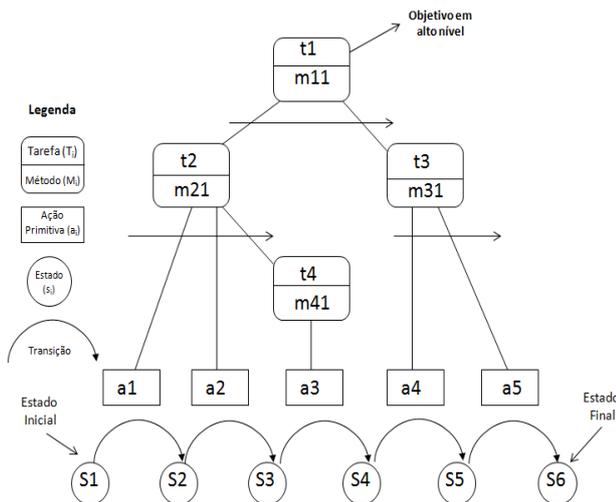


Fig. 4. Exemplo de um objetivo descrito como uma rede de tarefas hierárquicas (HTN) a partir de uma árvore de decomposição.

I. Planejamento sob Incerteza

Até o momento apresentaram-se modelos de planejamento cujas ações têm resultados determinísticos. Todavia, há casos em que pode ser útil supor que certas ações tenham mais de um resultado. É adequado, por exemplo, em situações onde o efeito de uma ação pode variar devido a mudanças aleatórias no ambiente ou a ações de outros agentes.

Na área espacial, tem sido explorado ainda de forma incipiente para ambientes dinâmicos, como os encontrados por sondas de exploração planetária. As técnicas de planejamento sob incerteza são baseadas em probabilidades e lidam com o não determinismo. Dentre elas, cita-se o *Markov Decision Process* (MDP) e o *Model Checking* (MC). Para exemplos na área espacial, veja respectivamente os trabalhos: [23], [53].

O MDP, embora seja poderoso e busque soluções ótimas, torna o espaço de busca do problema grande [50], [72] e tem sido aplicado a problemas menores, dada a dificuldade em

encontrar uma solução viável em termos de tempo e recursos computacionais [73]. A ideia do MC é resolver problemas de planejamento a partir de um modelo teórico. O domínio é modelado como um sistema de transição de estados não determinísticos, onde uma ação pode levar de um estado a vários estados diferentes. O objetivo é expresso por uma fórmula de lógica temporal [74].

Um dos entraves de usar esse tipo de planejamento é que para uma mesma entrada pode haver resultados distintos, o que dificulta os testes de robustez.

J. Formas de Representação Temporal em Sistemas de Planejamento e Escalonamento

A representação do tempo nos sistemas de planejamento e escalonamento é importante porque as ações e relações ocorrem ao longo do tempo. As informações temporais estabelecem quando os estados do domínio devem atingir os valores desejados, ou em que período determinadas tarefas devem ser realizadas. Há dois formalismos comumente aplicados: a álgebra temporal [75] e a álgebra de intervalo Allen [76], conhecida também como intervalo de Allen.

O primeiro opera sobre restrições binárias qualitativas entre instantes fixos, sem duração, denominados '*time points*' [8], com base nos seguintes símbolos de relação temporal: $\{<, >, =\}$. Nele, um ponto pode ser o início ou fim de algum evento, assim como um momento qualquer no tempo. O segundo opera sobre intervalos e restrições binárias qualitativas, utilizando treze relações primitivas: $\{Equal; Before; Meet; Overlap; FinishedBy; Contain; Start; During; StartBy; Finish; OverlapBy; MeetBy; After\}$ que pela relação inversa se reduz a sete, como mostra a Tabela IV.

No modelo de Allen, as ações e eventos passam a ocorrer dentro de períodos de tempo que possuem relacionamentos temporais entre si. As restrições binárias são utilizadas para estabelecer as relações temporais entre tarefas. Da mesma forma, são usadas por [75] para estabelecer as relações entre seus pontos no tempo. Sistemas de planejamento embarcado da área espacial têm adotado a representação proposta por Allen.

TABELA IV
RELAÇÕES TEMPORAIS DE ALLEN [76]

Relação	Símbolo	Inverso	Intervalo
$X \text{ equal } Y$	eq	eq	$\begin{array}{c} \text{-----}x \\ \text{-----}y \end{array}$
$X \text{ before } Y$	b	b_i	$\begin{array}{c} \text{-----}x \\ \text{-----}y \end{array}$
$X \text{ meets } Y$	m	m_i	$\begin{array}{c} \text{-----}x \\ \text{-----}y \end{array}$
$X \text{ overlaps } Y$	o	o_i	$\begin{array}{c} \text{-----}x \\ \text{-----}y \end{array}$
$X \text{ during } Y$	d	d_i	$\begin{array}{c} \text{-----}x \\ \text{-----}y \end{array}$
$X \text{ starts } Y$	s	s_i	$\begin{array}{c} \text{-----}x \\ \text{-----}y \end{array}$
$X \text{ finishes } Y$	f	f_i	$\begin{array}{c} \text{-----}x \\ \text{-----}y \end{array}$

V. PLANEJADORES EMBARCADOS EM SISTEMAS ESPACIAIS

Uma vez apresentadas as técnicas computacionais, podemos descrever os principais sistemas de planejamento embarcado.

A. RAX-PS e RAX

O RAX-PS, inspirado no *Heuristic Scheduling Testbed System* (HSTS [77]), manipula os objetivos na forma de um CSP. Trata-se de um planejador temporal baseado em *timelines* [78] e não utiliza a abordagem de HTN [35]. O mecanismo de planejamento utiliza uma variedade de algoritmos de propagação de restrições para localizar e resolver conflitos até que uma solução seja encontrada. Ele gera os planos que o RAX utiliza para controlar a sonda DS-1. O RAX-PS trata as tarefas de planejamento e escalonamento de forma unificada [13]. Uma limitação está no tempo de resposta para criar um novo plano devido à característica do planejamento em lote adotado.

B. CASPER e ASE

O CASPER, inspirado no ASPEN, utiliza o planejamento contínuo por meio da técnica de reparo iterativo [57]. Nela, os conflitos são detectados, em seguida, o algoritmo escolhe um conflito, e ao final, escolhe-se um método de reparo [79]. O processo é repetido até encontrar um plano sem falhas. Este planejador mantém um plano consistente com as informações atualizadas conforme o ciclo: mudanças no estado inicial ou dos objetivos; propagar os efeitos das mudanças; e invocar algoritmos de reparo para remoção dos conflitos. Após a operação autônoma do ASE [10] no EO-1, incrementou-se o retorno científico da missão em mais de cem vezes [3]. O CASPER também foi explorado em ambientes simulados para exploração planetária e testado em voo num *CubeSat* [80], [81].

C. EUROPA e IDEA

O EUROPA é um *framework* da NASA implementado em C++, descendente do HSTS, para representação e solução de problemas de satisfação de restrições com ênfase em redes de restrições temporais [82]. Este planejador provê interfaces para que um cliente possa modificar um plano, e consultá-lo quanto à sua viabilidade, consistência, decisões abertas, entre outros aspectos [83]. Ele foi aplicado a vários domínios, incluindo a operação em solo de missões interplanetárias e de constelação de satélites. Foi referência para construção de planejadores embarcados, como da arquitetura *Intelligent Distributed Execution Architecture* (IDEA [63]).

D. APSI e GOAC

O *Advanced Planning and Scheduling Initiative* (APSI [84]) é um *framework* de planejamento da ESA que utiliza a linguagem *Domain Definition Language* (DDL). Ele provê uma representação baseada em técnicas de IA para suportar sistemas de planejamento e escalonamento. Ao contrário de outros sistemas, não existe uma noção explícita de 'ação', a representação é baseada apenas em *timelines*. A arquitetura *Goal-Oriented Autonomous Controller* (GOAC [62]), voltada à maior autonomia de missões robóticas, desenvolveu seu planejador embarcado inspirado no APSI, porém adicionando a capacidade de replanejamento.

E. Letmedo e GOESA

O planejador embarcado *LetMeDo* foi utilizado na arquitetura *Goal-based Enabling Software Architecture* (GOESA [2]) do Instituto Nacional de Pesquisas Espaciais (INPE). Tal planejador deu continuidade a estudos anteriores

[85], [86] do INPE. Tem como objetivo ser de propósito geral e adequado à execução em bordo. O algoritmo para resolução do CSP é da classe de busca local, mas pode ser configurado também como de busca aleatória. A GOESA integra a solução de planejamento e escalonamento em um único algoritmo, que opera sobre uma linguagem de representação de domínio própria. Apresenta técnica similar à do reparo iterativo do CASPER. Emprega-se a modelagem por *timelines* e não se utiliza heurísticas de buscas.

F. Trabalhos Correlatos

Demais pesquisas para controlar um veículo espacial a partir de um planejador embarcado foram realizadas, porém aparentemente sem considerar um ambiente computacional compatível com o da área espacial. Em [4], [7], investiga-se planejamento distribuído por meio de uma arquitetura multiagentes. Os trabalhos ([4], [39], [71]) utilizam planejamento em HTN com estudo de caso voltado a satélites de sensoriamento remoto. Particularmente em [39], emprega-se o formalismo de Redes Petri para modelar diferentes abstrações da plataforma espacial. Em [87], apresenta-se uma técnica interessante para converter automaticamente o modelo PDDL na forma de restrições de tabela.

Outras abordagens ([88]-[90]) propõem maior autonomia através de técnicas que permitem o reparo em tempo real considerando computadores da área espacial, mas sem dispor de um planejador embarcado. Dentre elas, destaca-se o *Verification of Autonomous Mission Planning On-board Spacecraft* (VAMOS [90]) que combina a capacidade de replanejamento em tempo real com auxílio de sistemas de planejamento em solo que têm mais poder computacional.

VI. ANÁLISE COMPARATIVA ENTRE SISTEMAS DE PLANEJAMENTO EMBARCADO APLICADO A MISSÕES ESPACIAIS

As Tabelas V e VI apresentam um sumário comparativo das características essenciais dos planejadores. Embora os planejadores ASPEN, EUROPA e APSI sejam voltados originalmente para gerenciar operações de solo, eles tiveram versões adaptadas para serem embarcadas em aplicações espaciais. Nessa análise, foram consideradas as características de suas versões embarcadas à exceção do ASPEN.

TABELA V
ABORDAGENS E ESTRATÉGIAS DE PLANEJAMENTO

Nome do Planejador	Tipo de Planejamento	Estratégia	Linguagem
RAX-PS [9]	<i>Timeline</i> (CSP) Temporal	Planejamento em Lote (<i>Batch</i>)	PDDL
ASPEN [59]	HTN e CSP	Reparo Iterativo e Busca Construtiva	AML
CASPER [6]	<i>Timeline</i> (CSP) Temporal	Reparo Iterativo	Baseada na AML
EUROPA [60]	<i>Timeline</i> (CSP) Temporal	Reparo Iterativo	NDDL
APSI [84]	<i>Timeline</i> (CSP) Temporal	Reparo Iterativo	DDL
<i>LetMeDo</i> [2]	<i>Timeline</i> (CSP) Temporal	Similar ao Reparo Iterativo	ISIS

Por meio da Tabela V, nota-se que os planejadores com herança de voo e os que consideraram o ambiente real de operação de um satélite ficaram restritos a técnicas deterministas (CSP e HTN). Elas se mostraram viáveis para

execução a bordo de computadores espaciais, que possuem severas limitações de processamento e memória.

A maioria utiliza *timelines* para o gerenciamento de recursos aliado à álgebra de Allen para manipular restrições temporais, conforme as Tabelas V e VI. Observa-se pela Tabela VI, que os modelos embarcados são geralmente baseados em uma linguagem dependente de domínio. Isso traz melhores resultados de desempenho se comparado à abordagem independente de domínio [87].

TABELA VI
COMPARATIVO DAS CARACTERÍSTICAS DOS SISTEMAS DE PLANEJAMENTO

Planejador	RT	PL	HB	TL	CR	MPU	SPID
RAX-PS	IA	CE	Sim	Sim	Não	Não	Não
ASPEN	IA	CE	Sim	Sim	Não	Sim	Sim
CASPER	IA	CE	Sim	Sim	Sim	Não	Não
EUROPA	IA	DI	Sim	Sim	Sim	Sim	Sim
APSI	IA	DI	Sim	Sim	Sim	Sim	Sim
<i>LetMeDo</i>	TP	CE	Não	Sim	Sim	Sim	Não

RT = Representação Temporal, PL = Planejamento, HB = Heurísticas de Buscas, TL = *Timelines*, CR = Capacidade de Replanejamento, MPU = Modelo de Planejamento Unificado, SPID = Sistema de Planejamento Independente de Domínio, IA = Intervalo de Allen, TP = *Time Points*, CE = Centralizado, DI = Distribuído.

Com relação ao tempo de planejamento, ele depende dentre outros fatores, do tipo da missão, das características do problema, como tamanho do horizonte temporal, tipo de solicitação (criar um novo plano ou requisitar mudança em um plano preexistente), do grau de impacto das mudanças no plano original, e do estado dos equipamentos. Embora não se possam comparar diretamente os resultados de cada planejador, por tratarem de problemas distintos, modelados de formas específicas, traz-se aqui uma compilação dos dados de desempenho dos estudos que tiveram seus resultados divulgados. Os tempos e alocação do processador devem ser analisados à luz dos dados expostos na Tabela VII.

TABELA VII
RESULTADOS E TESTES DE DESEMPENHO

Nome	Domínio de Aplicação	Tempo de Planejamento	Algoritmo de Busca	Uso da CPU
RAX-PS	Científica Espaço Profundo (DS-1)	De 30 minutos a 4 horas dependendo do tamanho do problema	Busca em profundidade com retrocesso	50%
CASPER	Observação da Terra (EO-1)	Na ordem de dezenas de minutos	Busca Local	100%
<i>LetMeDo</i>	Observação da Terra (Amazonia-1)	Para novos planos, ou modificações de baixo impacto: 1 a 3 minutos. Pior caso: 25 minutos	Busca Local	100%

Sob o ponto de vista de desempenho, a estratégia que se mostrou mais satisfatória parece ser a abordagem de reparo iterativo, cuja ideia central é refinar o plano preexistente sem elaborar um novo plano completo. Outras melhorias podem ser alcançadas por heurísticas de buscas específicas (e.g., [56], [91]), ou aperfeiçoando a expressividade do modelo.

A Tabela VIII traz o ambiente computacional do estudo de caso ou ambiente real de voo dos trabalhos em que os

referidos dados de desempenho foram aferidos. Em virtude da severa limitação computacional de processadores qualificados, os algoritmos de planejamento embarcado têm empregado soluções incompletas e de menor custo computacional (e.g., busca local) em detrimento a soluções ótimas.

A próxima seção sumariza perspectivas e tendências futuras identificadas durante a pesquisa realizada neste trabalho para consolidar o estado da arte em planejamento automatizado voltado às missões espaciais.

TABELA VIII
AMBIENTE COMPUTACIONAL ESPACIAL

Planejador	Processador	Memória RAM	Tamanho da Imagem	RTOS
RAX-PS	PowerPC RAD6000 25 MHz	32 MB	-	VxWorks
CASPER	Mongoose V R3000 12 MHz 4 MIPS ERC32	40 MB para o CASPER	5 MB	VxWorks
<i>LetMeDo</i>	15 MHz 5.8 MIPS	4 MB	~ 100 kB	RTEMS

RTOS = *Real Time Operating System* e MIPS = *Millions of Instructions Per Second*.

VII. PERSPECTIVAS E TENDÊNCIAS FUTURAS DE PESQUISA

Como possibilidade de tendências futuras na área, as Tabelas IX e X apresentam assuntos ainda em fase incipiente de estudo e outros até agora não explorados no domínio espacial. A coluna mais à esquerda reúne os trabalhos de referência que já abordaram o tema.

TABELA IX
TENDÊNCIAS EM PLANEJAMENTO BASEADO EM OBJETIVOS NA ÁREA ESPACIAL

Ref.	Tema de Pesquisa	Motivação
[58]	Concepção de uma linguagem padronizada para o domínio embarcado da área espacial	Reuso da linguagem para concepção de novas aplicações de planejamento
[55], [87]	Estratégias para que o planejamento seja baseado em uma linguagem independente de domínio	Permite uso em outros domínios, facilitando a comparação entre os algoritmos
[23],[29] [53], [73]	Planejamento probabilístico para: (i) planejamento de missão e (ii) diagnóstico autônomo de falhas	Útil para lidar com o não determinismo do ambiente e com situações de falhas
[4]	Execução do plano por múltiplos processadores	Aumento de desempenho
[56]	Heurísticas para CSPs	Aumento de desempenho
[23]	Associar aprendizado de máquina aos modelos	Capacidade de adaptação do modelo ao ambiente real de operação
[25],[51], [52],[92]	Planejamento de missão em bordo distribuído por múltiplos satélites	Diminuir custo de operação para controlar vários satélites da missão

Destaca-se que algumas tendências da Tabela IX apostam na evolução tecnológica dos processadores espaciais. São exemplos disso: a pesquisa em planejadores baseados em linguagens independentes de domínio que possuem menor desempenho (vide [55], [87]) e o planejamento probabilístico que sofre com a explosão de estados ([50], [72], [93]).

Em áreas correlatas de pesquisa, há trabalhos interessantes baseados em problemas de otimização. Nessa linha, são tendências os problemas voltados ao escalonamento de imagens em satélites com autonomia em bordo ([11], [24], [94]), especialmente aqueles que utilizam como estudo de caso os satélites ágeis ([95]-[97]). Entretanto, a estrita restrição de tempo real raramente é considerada nesses estudos [95].

Dos temas ainda não explorados na área espacial (Tabela X), pode-se destacar o planejamento híbrido que combina as duas técnicas mais promissoras para aplicações espaciais e o uso da HTN considerando restrições temporais e de recurso.

TABELA X
TEMAS AINDA NÃO EXPLORADOS NO DOMÍNIO ESPACIAL

Ref.	Tema de Pesquisa	Motivação
[98],[99]	Modelo HTN com autoaprendizagem do domínio	Complexidade para o especialista descrever o domínio HTN
[49]	Planejamento em HTN com restrições temporais e de recursos	A maioria dos planejadores HTN não lida com restrições de tempo e de recursos
[78],[100]	Planejamento híbrido combinando técnicas entre planejamento hierárquico e temporal	Aliar o bom desempenho do planejador HTN com a capacidade para lidar com situações não planejadas do CSP
[101]	Mecanismos para reparo em HTN	Dificuldade do planejador HTN para lidar com falhas no plano
[102],[103]	Heurísticas para melhorar o desempenho do planejamento em HTN	Útil para missões espaciais de maior complexidade
[104]	Aprendizado de máquina em modelos de tomada de decisão	Modelo mais fidedigno ao contexto real de operação do satélite

VIII. CONCLUSÃO

O uso de técnicas de planejamento automatizado em aplicações reais permanece relativamente baixo, apesar de seu grande potencial [105]. De forma oposta a este cenário, uma das áreas de aplicação que vem obtendo sucesso experimental é justamente sua adoção em veículos espaciais com autonomia. A literatura registra apenas dois artigos de revisão ([3], [12]) neste assunto publicados em periódicos nas últimas duas décadas. Podemos elencar dois tópicos principais de pesquisa nessa área: (i) planejamento de missão em bordo e (ii) reconfiguração do satélite em caso de falhas. Eles quando integrados ao software de voo fazem uma plataforma espacial com alto grau de autonomia. O principal estudo de revisão [12] até o momento visou se aprofundar na temática (ii). Os autores não encontraram trabalhos que tenham se concentrado no tema (i). Com objetivo de contribuir nessa área, o presente artigo fez um estudo direcionado à autonomia de veículos espaciais. Foi dada ênfase ao planejamento de missão baseado em objetivos, abordando as principais técnicas e modelos de planejamento em IA usados para controlar um satélite com autonomia em bordo. Um estudo comparativo dos trabalhos foi realizado, além de delinear novos rumos de pesquisas futuras. A principal contribuição deste trabalho é estabelecer o estado da arte em planejamento automatizado para o domínio embarcado da área espacial cujo tema foi pouco explorado pelos estudos atuais.

REFERÊNCIAS

- [1] M. Bennett, D. Dvorak, J. Hutcherson, M. Ingham, R. Rasmussen and D. Wagner, "An Architectural Pattern for Goal-Based Control," in *Proc. IEEE Aerospace Conference*, 2008, pp. 1-17.
- [2] F. N. Kucinskis, and M. G. V. Ferreira, "Planning on-board satellites for the goal-based operations for space missions," *IEEE Latin America Transactions*, vol. 11, no. 4, p. 1110-1120, Jun. 2013.
- [3] A. K. Jonsson, R. A. Morris and L. Pedersen, "Autonomy in space exploration: current capabilities and future challenges," *AI Magazine*, vol. 28, no. 4, pp. 27-42, 2007.
- [4] F. Amigoni, S. Gualandi, D. Menotti, and G. Sangiovanni, "A multiagent architecture for controlling the Palamede satellite," *Web Intelligence and Agent Systems*, vol. 8, no. 3, pp. 269-289, 2010.
- [5] S. Lemai, X. Olive and M. C. Charneau, "Decisional architecture for autonomous space systems," in *Proc. 9th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, 2006.
- [6] S. Knight, G. Rabideau, S. Chien, B. Engelhardt and R. Sherwood, "Casper: space exploration through continuous planning," *IEEE Intelligent Systems*, vol. 16, no. 5, pp. 70-75, Set. 2001.
- [7] X. Rui, C. Pinyuana, and X. Xiaofeib, "Realization of multi-agent planning system for autonomous spacecraft," *Advances in Engineering Software*, vol. 36, no. 4, pp. 266-272, Apr. 2005.
- [8] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning - Theory and Practice*, 1st ed., San Francisco, USA : Elsevier, 2004.
- [9] N. Muscettola, P. Nayak, B. Pell, and B. C. Williams, "Remote Agent: to boldly go where no AI system has gone before," *Artificial Intelligence*, vol. 103, no. 1, pp. 5-47, 1998.
- [10] S. Chien, R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, and A. Davis, "Using Autonomy Flight Software to Improve Science Return on Earth Observing One (EO-1)," *Journal of Aerospace Computing, Information and Communication*, vol. 2, no. 4, pp. 196-216, Apr. 2005.
- [11] Y. He, Y. Wang, Y. Chen, Y. and L. Xing, "Auto Mission Planning System Design for Imaging Satellites and Its Applications in Environmental Field," *Polish Maritime Research*, vol. 23, 2016.
- [12] M. Tipaldi, and L. Glielmo, "A Survey on Model-Based Mission Planning and Execution for Autonomous Spacecraft," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3893-3905, Dec. 2018.
- [13] K. Rajan, D. Bernard, G. Dorais, E. Gamble, B. Kanefsky, J. Kurien, W. Millar, N. Muscettola, P. Nayak, N. Rouquette, B. Smith, W. Taylor, and Y. Tung, "Remote Agent: An Autonomous Control System for the New Millennium," in *Proc. 14th European Conference on Artificial Intelligence*, 2000, pp. 726-730.
- [14] J. Vago, F. Westall, Pasteur Teams, Landing Site Selection Working Group, and Other Contributors, "Habitability on early Mars and the search for biosignatures with the ExoMars Rover," *Astrobiology*, vol. 17, no. 6 and 7, pp. 471-510, 2017.
- [15] T. V. Peters, J. Branco, D. Escorial, L. T. Castellani, A. Cropp, "Mission analysis for PROBA-3 nominal operations," *Acta Astron.*, vol. 102, pp. 296-310, Oct. 2014.
- [16] S. Chien, and R. Morris, "Space Applications of Artificial Intelligence," *AI Magazine*, vol. 35, no. 4, 2014.
- [17] K. Kolcio, L. Breger and P. Zetocha, "Model-Based Fault Management for Spacecraft Autonomy," in *Proc. IEEE Aerospace Conference*, 2014, pp. 1-14.
- [18] M. Esteve, J. Katoen, V. Y. Nguyen, B. Postma, and Y. Yushstein, "Formal correctness, safety, dependability, and performance analysis of a satellite," in *Proc. 34th International Conference on Software Engineering*, 2012, pp. 1022-1031.
- [19] W. Trzuskowski, L. Hallock, C. Rouff, J. Karlin, J. Rash, M. Hinchey, and R. Sterritt, *Autonomous and autonomic systems: with applications to NASA intelligent spacecraft operations and exploration systems*, 1st ed., New York, USA : Springer-Verlag, 2009.
- [20] T. Grant, A. O. Soler, A. Bos, U. Brauer, M. Neerinx and M. Wolff, "Space autonomy as migration of functionality: the Mars case," in *Proc. 2nd IEEE International Conference on Space Mission Challenges for Information Technology*, 2006, pp. 201-207.
- [21] A. Wander, and R. Forstner, "Innovative Fault Detection, Isolation and Recovery on-board Spacecraft: Study and Implementation using Cognitive Automation," in *Proc. 2nd International Conference on Control and Fault Tolerant Systems*, 2013, pp. 336-341.
- [22] C. Frost, "Challenges and opportunities for autonomous systems in space," in *Proc. National Academy of Engineering U.S. Frontiers of Engineering Symposium*, 2010.

- [23] G. D'angelo, M. Tipaldi, L. Glielmo and S. Rampone, "Spacecraft autonomy modeled via Markov decision process and associative rule-based machine learning," in *Proc. IEEE International Workshop on Metrology for AeroSpace*, 2017, pp. 324-329.
- [24] R. Axmann, "Interactive acquisition scheduling for low earth orbiting satellites," Ph.D dissertation, Abteilung für Maschinenbau, Technische Universität München, Munich, Germany, 2010.
- [25] S. Grey, "Distributed agents for autonomous spacecraft," Ph.D dissertation, Philosophy School of Engineering College of Science, Engineering University of Glasgow, United Kingdom (UK), 2013.
- [26] J. Wertz, and W. Larson, *Space mission analysis and design*, 3rd ed., Torrance, USA: Microcosm, Inc. and Kluwer Academic Publishers, 1999.
- [27] M. Koller, V. Reggestad, and K. Adamson, "ESOC Earth Observation Missions and the Automation of Operational Routine Tasks," in *Proc. 11th International Conference on Space Operations*, 2010.
- [28] M. Tipaldi, and L. Glielmo, "A Markovian based approach for autonomous space systems," in *Proc. IEEE Metrology for Aerospace*, 2015, pp. 426-430.
- [29] V. Nardone, A. Santone, M. Tipaldi, and L. Glielmo, "Probabilistic model checking applied to autonomous spacecraft reconfiguration," in *Proc. IEEE Metrology for Aerospace*, 2016, pp. 556-560.
- [30] J. Eickhoff, *Onboard computers, onboard software and satellite operations: An introduction*, Heidelberg, Germany: Springer-Verlag, 2012.
- [31] S. Bensalem, K. Havelund and A. Orlandini, "Verification and validation meet planning and scheduling," *Int. Journal on Software Tools for Technology Transfer*, vol. 16, no. 1, pp. 1-12, Feb. 2014.
- [32] M. Bozzano, A. Cimatti, M. Roveri, and A. Tchaltev, "A Comprehensive Approach to On-Board Autonomy Verification and Validation," in *Proc. 22th International Joint Conferences on Artificial Intelligence*, 2011.
- [33] P. Muñoz, A. Cesta, A. Orlandini and M. D. R-Moreno, "First Steps on an On-Ground Autonomy Test Environment," in *Proc. IEEE International Conference on Space Mission Challenges for Information Technology*, 2014, pp. 30-37.
- [34] F. N. Kucinskis, "Validation of Reasoning and Decision-Making of a Satellite Autonomous On-board Software," in *Proc. Fifth Latin-American Symposium on Dependable Computing*, 2011.
- [35] S. Chien, B. Smith, G. Rabideau, N. Muscettola and K. Rajan, "Automated planning and scheduling for goal based autonomous spacecraft," *IEEE Intelligent Systems*, vol. 13, no. 5, pp. 50-55. Sept-Oct. 1998.
- [36] E. Vashev, and M. Hinchey, "On the autonomy requirements for space missions," in *Proc. 16th IEEE Int. Sym. on Object/component/service-oriented Real-time distributed Computing*, 2013, pp 1-10.
- [37] S. Chien, D. Tran, G. Rabideau, S. Schaffer, D. Mandl, and S. Frye, "Improving the operations of the Earth Observing One mission via automated mission planning," in *Proc. 11th International Conference on Space Operations*, 2010.
- [38] Y. Gao, N. Policella and F. Kirchner, "Computational Intelligence for Space Systems and Operations," *IEEE Computational Intelligence Magazine*, vol. 8, no. 4, pp. 10-62, Nov. 2013.
- [39] A. Indra, V. K. Agrawal, and V. V. S. Sarma, "Stratified agent architecture for on-board mission planning and execution for an autonomous spacecraft," in *Proc. IEEE Region TENCON 10 Conf.*, 2008, pp. 1-6.
- [40] L. Pan, "Autonomous Rock Science Analysis System For Planetary Exploration," Ph.D dissertation, Department of Computer Science, Aberystwyth University, Wales, United Kingdom (UK), 2016.
- [41] European Committee for Space Standardization, *Space engineering - Space segment operability*, ECSS-E-ST-70-11C, 2008.
- [42] M. Tipaldi, M. Witzmann, M. Ferraguto, and L. Glielmo, "An approach for geostationary satellite mode management," in *Proc. IFAC-PapersOnLine*, vol. 50, no. 1, 2017, pp. 7241-7246.
- [43] X. Olive, "FDI(R) for satellites: How to deal with high availability and robustness in the space domain?," *Int. Journal of Applied Mathematics and Computer Science*, vol. 22, no. 1, pp. 99-107, Mar. 2012.
- [44] European Committee for Space Standardization, *Space Engineering - Spacecraft On-Board Control Procedures*, ECSS-E-ST-70-01C, 2010.
- [45] D. Dvorak, M. D. Ingham, and J. R. Morris, "Goal-based operations: an overview," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, Mar. 2009.
- [46] R. Francis, T. Estlin, D. Gaines, B. Bornstein, S. Schaffer, V. Verma, R. Anderson, M. Burl, S. Chu, and R. Castaño et al., "AEGIS autonomous targeting for the Curiosity rover's ChemCam instrument," in *Proc. IEEE Applied Imagery Pattern Recognition Workshop*, 2015, pp. 1-5.
- [47] S. Chien, D. Silverman, A. G. Davies and D. Mandl, "Onboard Science Processing Concepts for the HypIRI Mission," *IEEE Intelligent Systems*, vol. 24, no. 6, pp. 12-19, Nov.-Dec. 2009.
- [48] N. C. Freitas, P. P. Reboucas, C. D. G. Moura and M. P. S. Silva, "AgentGeo: Multi-Agent System of Satellite Images Mining," *IEEE Latin America Transactions*, vol. 14, no. 3, pp. 1343-1351, Mar. 2016.
- [49] C. Qi et al., "Hierarchical task network planning with resources and temporal constraints," *Knowledge-Based Systems*, vol. 133, pp. 17-32, 2017.
- [50] D. E. Smith, J. Frank, and A. K. Jonsson, "Bridging the gap between planning and scheduling," *Knowledge Engineering Review*, vol. 15, no. 1, pp. 47-83, 2000.
- [51] Z. Zheng, J. Guo, and E. Gill, "Onboard autonomous mission re-planning for multi-satellite system," *Acta Astronautica*, vol. 145, pp. 28-43, 2018.
- [52] Z. Zheng, J. Guo, and E. Gill, "Distributed onboard mission planning for multi-satellite systems," *Aerospace Science and Technology*, vol. 89, pp. 111-122, 2019.
- [53] M. Bozzano, A. Cimatti, A. Guiotto, A. Martelli, M. Roveri, A. Tchaltev, and Y. Yushstein, "On-board autonomy via symbolic model-based reasoning," in *Proc. 10th ESA Workshop on Advanced Space Technologies for Robotics And Automation*, 2008, pp. 1-8.
- [54] J. C. González, J. C. Pulido and F. Fernández, "A three-layer planning architecture for the autonomous control of rehabilitation therapies based on social robots," *Cognitive Systems Research*, vol. 43, pp. 232-249, 2017.
- [55] T. Nogueira, S. Fratini and K. Schilling, "Autonomously controlling flexible timelines: From domain-independent planning to robust execution," in *Proc. IEEE Aerospace Conference*, 2017, pp. 1-15.
- [56] H. Bu, J. Zhang and Y. Luo, "Constraint satisfaction and optimization for space station short-term mission planning based on an iterative conflict-repair method," *Journal of Engineering Optimization*, vol. 48, Feb. 2016.
- [57] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using iterative repair to increase the responsiveness of planning and scheduling for autonomous spacecraft," in *Proc. International Joint Conference on Artificial Intelligence*, 1999.
- [58] S. Fratini, N. Policella, and A. Donati, "Pattern-Based Modeling for Timeline Planning in Space Domains," *Acta Futura*, vol. 9, pp. 73-82, 2014.
- [59] R. Knight, C. Chouinard, G. Jones, and D. Tran, "Leveraging multiple artificial intelligence techniques to improve the responsiveness in operations planning: Aspen for orbital express," *AI Magazine*, vol. 35, no. 4, pp. 26-36, 2014.
- [60] J. Barreiro, M. Boyce, M. Do, J. Frank, M. Iatauro, T. Kichkaylo, P. Morris, J. Ong, E. Remolina, T. Smith, and D. Smith, "EUROPA: a platform for AI planning, scheduling, constraint programming, and optimization," in *Proc. 4th International Competition on Knowledge Engineering for Planning and Scheduling*, 2012.
- [61] F. N. Kucinskis, and M. G. V Ferreira, "Taking the ECSS Autonomy Concepts One Step Further," in *Proc. 11th International Conference on Space Operations*, 2010, p. 187-199.
- [62] A. Ceballos, S. Bensalem, A. Cesta, L. Silva, S. Fratini, F. Ingrand, J. Ocón, A. Orlandini, K. Rajan, R. Rasconi, and M. Winnendael, "A goal-oriented autonomous controller for space exploration," in *Proc. 11th Symposium on Advanced Space Technologies in Robotics and Automation*, 2011.
- [63] N. Muscettola, G. Dorais, C. Fry, R. Levinson, and C. Plaunt, "IDEA: planning at the core of autonomous reactive agents," in *Proc. International Conf. on AI Planning and Scheduling*, 2002, pp. 58-60.
- [64] M. Tipaldi, C. Legendre, O. Koopmann, M. Ferraguto, R. Wenker, and G. D'Angelo, "Development strategies for the satellite flight software on-board Meteosat Third Generation," *Acta Astronautica*, vol. 145, pp. 482-491, 2018.
- [65] H. Wojtkowiak, O. Balagurin, G. Fellingner and H. Kayal, "ASAP: Autonomy through on-board planning," in *Proc. 6th Int. Conf. on Recent Advances in Space Technologies*, 2013, pp. 377-381.
- [66] J. M. Victoria, "Automated Hierarchical, Forward-Chaining Temporal Planner for Planetary Robots Exploring Unknown Environments," M.S. thesis, Vom Fachbereich Informatik, Technische Universität Darmstadt, Darmstadt, Germany, 2016.
- [67] M. C. Mayer, A. Orlandini, and A. Umbrico, "Planning and execution with flexible timelines: A formal account," *Acta Informatica*, vol. 53, no. 6-8, pp. 649-680, 2016.

- [68] S. Fratini, F. Pecora, and A. Cesta, "Unifying Planning and Scheduling as Timelines in a Component-Based Perspective," *Archives of Control Sciences*, vol. 18, no. 2, pp. 231-271, 2008.
- [69] S. Chien, M. Johnston, J. Frank, and M. Giuliano et al., "A generalized timeline representation, services, and interface for automating space mission operations," in *Proc. 11th Int. Conf. on Space Operations*, 2012.
- [70] I. Georgievski, and M. Aiello, "HTN planning: Overview, comparison, and beyond," *Artificial Intelligence*, vol. 222, pp. 124-156, May. 2015.
- [71] Z. Zhang, R. He, and Y. Tan, "An HTN-based mission planning model and algorithm for autonomous spacecraft," in *Proc. 1st Int. Sym. on Systems and Control in Aerospace and Astronautics*, 2006, pp. 4-26.
- [72] S. Lemai, "IXTET-EXEC: planning, plan repair and execution control with time and resource management," Ph.D dissertation, Institut National Polytechnique de Toulouse (INPT), Toulouse, France, 2004.
- [73] D. Codetta-Raiteri and L. Portinale, "Dynamic Bayesian Networks for Fault Detection, Identification, and Recovery in Autonomous Spacecraft," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 13-24, Jan. 2015.
- [74] V. Nardone A. Santone, M. Tipaldi, D. Liuzza and L. Glielmo, "Model checking techniques applied to satellite operational mode management," *IEEE Systems Journal*, pp. 1-12, 2018.
- [75] M. B. Vilain, and H. Kautz, "Constraint propagation algorithms for temporal reasoning," in *Proc. fifth American Institute of Aeronautics and Astronautics*, 1986, pp. 377-382.
- [76] J. Allen, "Maintaining knowledge about temporal intervals," *Communications of the Association for Computing Machinery*, vol. 26, no. 11, pp. 832-843, 1983.
- [77] N. Muscettola, "HSTS: Integrated planning and scheduling," *Intelligent Scheduling*, Morgan Kaufman, pp. 169-212, 1994.
- [78] F. Dvorak, R. Barták, A. Bit-Monnot, F. Ingrand and M. Ghallab, "Planning and Acting with Temporal and Hierarchical Decomposition Models," in *Proc. IEEE International Conference on Tools with Artificial Intelligence*, 2014, pp. 115-121.
- [79] D. Tran, S. Chien, G. Rabideau, and B. Cichy, "Flight software issues in onboard automated planning - lessons learned on EO-1," in *Proc. International Workshop on Planning and Scheduling for Space*, 2004.
- [80] S. Chien, J. Doubleday, D. Thompson, K. Wagstaff, J. Bellardo, C. Francis, E. Baumgarten, A. Williams, E. Yee, E. Stanton, and J. Piug-Suari, "Onboard Autonomy on the Intelligent Payload Experiment CubeSat Mission," *J. Aerospace Inf. Sys.*, vol. 14, pp. 307-315, 2017.
- [81] A. Altinok, D. R. Thompson, B. Bornstein, S. Chien, J. Doubleday, and J. Bellardo, "Real-Time Orbital Image Analysis Using Decision Forests, with a Deployment Onboard the IPEX Spacecraft," *J. Field Robotics*, vol. 33, pp. 187-204, 2015.
- [82] P. Aschwanden, V. Baskaran, S. Bernardini, C. Fry, M. Moreno, N. Muscettola, C. Plaunt, D. Rijsman, and P. Tompkins, "Model-unified planning and execution for distributed autonomous system control," in *Proc. Workshop on spacecraft autonomy: using AI to expand human space exploration*, 2006.
- [83] J. Frank and A. Jonsson, "Constraint-based attribute and interval planning," *Journal of Constraints Special Issue on Constraints and Planning*, vol. 8, no. 4, pp. 339-364, 2003.
- [84] S. Fratini, and A. Cesta, "The APSI Framework: A Platform for Timeline Synthesis," in *Proc. 1st Workshops on Planning and Scheduling with Timelines*, 2012, pp. 8-15.
- [85] F. N. Kucinskis, and M. G. V. Ferreira, "Dynamic Allocation of Resources to Improve Scientific Return with Onboard Automated Planning," in *Proc. 9th Int. Conf. on Space Operations*, 2006.
- [86] P. E. Cardoso and M. G. V. Ferreira, "Proposal of a New Architecture for Using the Dynamic Object Model," *IEEE Latin America Transactions*, vol. 5, no. 3, pp. 177-184, Jun. 2007.
- [87] X. Jiang and R. Xu, "A Constraint-Programmed Planner for Deep Space Exploration Problems With Table Constraints," *IEEE Access*, vol. 5, pp. 17258-17270, 2017.
- [88] M. J. Woods, L. Baldwin, G. Wilson, S. Hall, and A. Pidgeon, "MMOPS: assessing the impact of on-board autonomy for planetary exploration missions," in *Proc. 9th Int. Conf. on Space Operations*, 2006.
- [89] B. Wille, M. T. Wörle, C. Lenzen, "VAMOS - Verification of Autonomous Mission Planning On-board a Spacecraft," in *Proc. 9th Symposium on Automatic Control in Aerospace*, vol. 46, no. 19, 2013, pp. 382-387.
- [90] C. Lenzen, M. T. Woerle, T. Göttfert, F. Mrowka, and M. Wickler, "Onboard Planning and Scheduling Autonomy within the Scope of the FireBird Mission," in *Proc. 13th International Conference on Space Operations*, 2014.
- [91] X. Jiang, P. Cui, R. Xu, A. Gao and S. Zhu, "An action guided constraint satisfaction technique for planning problem," in *Proc. IEEE 15th International Conference on Cognitive Informatics & Cognitive Computing*, 2016, pp. 167-173.
- [92] T. Nogueira, S. Fratini, and K. Schilling, "Planning and Execution to Support Goal-based Operations for NetSat: a Study," in *Proc. 10th International Workshop on Planning and Scheduling for Space*, 2017.
- [93] S. A. Jacklin, "Survey of verification and validation techniques for small satellite software development," in *Proc. Space Tech Expo Conf.*, 2015, pp. 1-20.
- [94] S. Peng, H. Chen, C. Du, J. Li and N. Jing, "Onboard Observation Task Planning for an Autonomous Earth Observation Satellite Using Long Short-Term Memory," *IEEE Access*, vol. 6, pp. 65118-65129, 2018.
- [95] Y. She, S. Li, and Y. Zhao, "Onboard mission planning for agile satellite using modified mixed-integer linear programming," *Aerospace Science and Technology*, vol. 72, pp. 204-216, 2018.
- [96] X. Chu, Y. Chen, and Y. Tan, "An anytime branch and bound algorithm for agile earth observation satellite onboard scheduling," *Advances in Space Research*, vol. 60, no. 9, pp. 2077-2090, 2017.
- [97] Y. Song, D. Huang, Z. Zhou, and Y. Chen, "An emergency task autonomous planning method of agile imaging satellite," *Journal on Image and Video Proc.* vol. 29, Apr. 2018.
- [98] H. Zhuo, H. Muñoz Avila, Q. Yang, "Learning hierarchical task network domains from partially observed plan traces," *Artif. Intell.*, vol. 212, pp. 134-157, 2014.
- [99] C. Nguyen, N. Reifsnnyder, S. Gopalakrishnan and H. Munoz-Avila, "Automated learning of hierarchical task networks for controlling minecraft agents," in *Proc. IEEE Conference on Computational Intelligence and Games*, 2017, pp. 226-231.
- [100] J. Victoria, S. Fratini, N. Policella, O. Stryk, Y. Gao, and A. Donati, "Planning Mars Rovers with Hierarchical Timeline Networks," *Acta Futura* vol. 9, pp. 21-29, 2014.
- [101] D. Holler, P. Bercher, G. Behnke, and S. Biundo, "HTN Plan Repair Using Unmodified Planning Systems," in *Proc. 1th International Workshop on Planning and Scheduling for Space Workshop on Hierarchical Planning*, 2018, pp. 26-30.
- [102] K. Cheng, L. Wu, X. Yu, C. Yin, and R. Kang, "Improving hierarchical task network planning performance by the use of domain-independent heuristic search," *Knowledge-Based Systems*, vol. 142, pp. 117-126, 2018.
- [103] D. Höller, P. Bercher, G. Behnke, and S. Biundo, "A generic method to guide HTN progression search with classical heuristics," in *Proc. Int. Conf. on Automated Planning and Scheduling*, 2018, pp. 114-122.
- [104] G. D'Angelo, S. Rampone, and F. Palmieri, "Developing a trust model for pervasive computing based on apriori association rules learning and bayesian classification," *Soft Computing*, vol. 21, pp. 6297-6315, 2017.
- [105] M. Ghallab, D. Nau, and P. Traverso, "The actor's view of automated planning and acting: A position paper," *Artificial Intelligence*, vol. 208, pp. 1-17, 2014.



Filipe De Simone Cividanes tem graduação em Ciência da Computação pela Universidade Estadual Paulista (UNESP, 2007). Obteve o título de mestre em Engenharia Eletrônica e Computação pelo Instituto Tecnológico de Aeronáutica (ITA, 2010). É atualmente doutorando do Instituto Nacional de Pesquisas Espaciais (INPE) na área de Engenharia e Tecnologia Espaciais. Como Tecnologista pleno do INPE da Divisão de Sistemas de Solo, tem experiência na área espacial com software de tempo real para computadores de bordo de satélites. Atualmente, está atuando no desenvolvimento do software embarcado e também do emulador do computador de bordo da missão Amazonia-1 do INPE.



Maurício Gonçalves Vieira Ferreira é Doutor em Ciência da Computação pelo INPE na área de Engenharia de Software desde 2001. Em 1996, tornou-se Mestre em Ciência da Computação na área de Engenharia de Software novamente pelo INPE. Como Tecnologista sênior, trabalha no Centro de Controle de Satélites no INPE desde 1988, suportando o time de

desenvolvimento de software de controles de satélite. Na área Acadêmica, é professor titular do curso de pós-graduação do INPE. Desde 2001, lidera estudantes de mestrado e doutorado. Produziu mais de 150 artigos, orientou 14 doutores e 23 mestres. Também é membro do SPACEOPS - *Technical Program Committee*.



Fabrício de Novaes Kucinskis tem graduação em Engenharia de Computação pela Universidade de Mogi das Cruzes (2003), mestrado em Computação Aplicada pelo INPE (2007) e doutorado em Engenharia e Tecnologia Espaciais também pelo INPE (2012). Como Tecnologista sênior do Grupo de Supervisão de Bordo de Satélites, da

Divisão de Eletrônica Aeroespacial do INPE, está atualmente coordenando e desenvolvendo o software embarcado para a missão Amazonia-1. Participou de processos de transferência de tecnologia em controle de atitude e órbita na empresa argentina INVAP SE e em plataformas para satélites geoestacionários de telecomunicações na ítalo-francesa *Thales Alenia Space*.