Proposta de Método de Aprendizado de Modelo de Ação para Planejamento em Domínios Incompletos

Romulo de Oliveira Leite
Volmir Eugenio Wilhelm
Programa de Pós-Graduação em Métodos Numéricos em Engenharia
Universidade Federal do Paraná
Curitiba, Brasil
romulool@yahoo.com.br, volmirw@gmail.com

***Resumo*—Este trabalho apresenta uma proposta de método de aprendizagem de modelos de ação baseado em casos, aplicado à solução de problemas de planejamento sobre modelos de domínios STRIPS incompletos. Um experimento preliminar sobre o domínio 'Mundo dos Blocos' mostra a efetividade do método proposto na prospecção de características do domínio em um banco de casos corretamente solucionados.**

***Palavras-chave*—planejamento automatizado; aprendizagem de modelo de ação; domínios incompletos**

I. Introdução

O problema de planejamento automatizado sobre modelos de domínios incompletos em representação STRIPS (*STanford Research Institute Problem Solver*) consiste na criação de planos que sejam soluções para problemas formulados sobre determinado domínio ainda que os conjuntos de precondições e efeitos de seus operadores não sejam completamente conhecidos ou declarados. Esse problema surge da dificuldade inerente à construção de modelos completos, a qual usualmente é uma tarefa trabalhosa e não-trivial, além de estar sujeita a erros [1].

A maioria das técnicas desenvolvidas para o planejamento automatizado requer a especificação completa do domínio. Por conta disso, tais técnicas apresentam deficiência de robustez quando aplicadas a modelos incompletos, isto é, perdem acurácia na criação de planos que sejam soluções efetivas em um contexto completo. Trabalhos como os de [2], [3] e [4] sugerem que essa dependência de completude representa um gargalo na aplicação de novas tecnologias de planejamento.

Quando não é possível garantir a completude do modelo é necessário que haja uma fonte suplementar de informação para prover conhecimento a respeito da dinâmica do domínio. No planejamento baseado em casos (*Case-Based Planning*), essa fonte de informação consiste em um conjunto de problemas de planejamento cuja solução seja conhecida para o modelo completo.

Para lidar com o problema de planejamento em domínios incompletos, o presente trabalho apresenta a proposta de um método que possibilite a formulação ou o refinamento de um modelo STRIPS. Tal método se caracteriza pela inflação dos conjuntos dos operadores e a posterior redução desses conjuntos com base na informação fornecida pelo conjunto de casos.

II. Definição do Problema

Seja L uma linguagem de primeira ordem sem funções definidas e com um número finito de símbolos de predicados e de constantes; um domínio STRIPS sobre L é definido como um sistema estado-transição $\Sigma = (S, A, \gamma)$, em que S é um conjunto finito de estados STRIPS, A é um conjunto de instâncias de operadores STRIPS e γ é a função de transição.

Seja O o conjunto de operadores em Σ; um operador $o \in O$ é definido como uma tripla (nome(o), pre(o), ef(o)), em que nome(o) é o nome do operador, pre(o) é o conjunto de precondições de o e ef(o) é o conjunto de efeitos de o. Por sua vez, ef(o) possui dois subconjuntos disjuntos: $ef^{+}(o)$ é o conjunto de efeitos positivos (*add list*) e $ef^{-}(o)$ é o conjunto de efeitos negativos (*delete list*).

Essencialmente, um domínio STRIPS é descrito pelos seus operadores; assim, a completude de um domínio depende da completude de cada operador. Um operador é dito incompleto se há algum literal omitido de seus subconjuntos. Em consequência, um domínio que contenha algum operador incompleto é chamdo de 'domínio incompleto'.

Um problema de planejamento P é uma tripla (Σ, s_0, g) em que Σ é um domínio STRIPS, $s_0 \in S$ é o estado inicial e g é uma meta. Uma solução para um problema de planejamento é um plano $\pi = \{ a_1, a_2, \ldots, a_k \}$ em que $a_1, a_2, \ldots, a_k$ é uma sequência ordenada de instâncias dos operadores de O, ou seja, $a_i \in A$, para i = 1, ..., k.

Dessa forma, dado um domínio incompleto Σ^*, um problema $P = (\Sigma^*, s_0, g)$ e um conjunto C de problemas resolvidos corretamente com respeito ao domínio completo Σ, o problema de planejamneto em domínios incompletos consiste em encontrar uma solução para P que seja correta com respeito a Σ [5].

III. O Método Proposto

O algoritmo do método proposto é composto de duas fases. A primeira tem por objetivo inflar os conjuntos de

precondições e efeitos de todos os operadores do domínio, completando-os com todos os literais que são candidatos a integrarem o modelo. Em um primeiro passo, os conjuntos de efeitos de cada operador são preenchidos e, em um segundo passo, cada solução dos casos pertencentes ao conjunto de planos corretos é usada para guiar a inflação dos conjuntos de precondições. Essa primeira fase é chamada de 'Fase de Inflação'.

O modelo a ser processado pelo algoritmo pode ser tanto um modelo vazio — o que equivale à formulação plena do modelo — quanto um esboço do modelo completo no qual estejam faltando literais. Neste caso, todos os literais fornecidos devem ser corretos. Às precondições e aos efeitos fornecidos dá-se o nome de 'precondições e efeitos fixos', enquanto que aos literais adicionados pelo algoritmo são dá-se o nome de 'precondições e efeitos inflados'.

Na segunda fase, chamada de 'Fase de Redução', uma sequência de filtragens retira do modelo os literais que não forem compatíveis com os casos do conjunto de planos solucionados.

A primeira redução impõe restrições a cada operador individualmente. A intersecção entre o conjuntos de precondições e de efeitos positivos de um operador deve ser vazia; assim, o algoritmo retira do modelo todos os literais que violam esta restrição.

A segunda redução faz uso do conjunto de planos corretos para eliminar inconsistências no modelo. A Fase de Inflação pode levar o modelo a conter os literais l e ¬l em um mesmo conjunto; se um destes for inflado e estiver em conflito com a ação precedente (no caso de uma precondição) ou subsequente (no caso de um efeito), o algoritmo o retirará do modelo.

A terceira redução elimina todas as inconsistências que eventualmente permaneçam dentro de cada subconjunto de todos os operadores. Se após a primeira e a segunda reduções existirem literais l e ¬l dentro de algum desses subconjuntos, o algoritmo providenciará a remoção de ambos.

A quarta redução faz uso dos axiomas do domínio para retirar eventuais precondições redundantes; isto é, caso alguma precondição inflada seja implicada por uma precondição fixa, o algortimo removerá essa precondição inflada do operador. Esse procedimento tem por objetivo minimizar as precondições negativas.

IV. Resultados Preliminares

O método proposto foi testado em problemas formulados sobre o domínio Mundo dos Blocos (*Blocks World*). Esses testes foram conduzidos vanriando-se os seguintes parâmetros: quantidade de blocos no conjunto de casos corretos fornecidos (5, 10 ou 15), número de casos corretos fornecidos (5, 10 ou 30) e completude inicial do modelo (0%, 25%, 50%, 75% ou 100%). Ao todo, 450 problemas foram resolvidos por meio do algoritmo SATPLAN06 [6].

Um resultado que merece destaque é a relação observada entre a completude inicial e as proporções de soluções corretas, de soluções falsas e de declarações incorretas de infactibilidade. À medida que a completude inicial é maior, a proporção de soluções corretas cresce, enquanto que a de infactibilidades decresce. No entanto, o comportamento da proporção de soluções falsas não apresenta comportamento monótono: a incidência desse tipo de solução aumenta à medida que a completude vai de 0% a 100%, sem todavia atingir os 100%. Dos modelos que alcançaram 100% de completude ou que ultrapassaram esse valor após o processamento (com a presença de literais incorretos), nenhum forneceu solução falsa, o que caracteriza um bom resultado, dado que assume-se por hipótese que esse tipo de erro tem potencial de prejuízo superior às infactibilidades. O gráfico da Fig. 1 exibe a relação entre a completude inicial e os tipos de soluções obtidas.

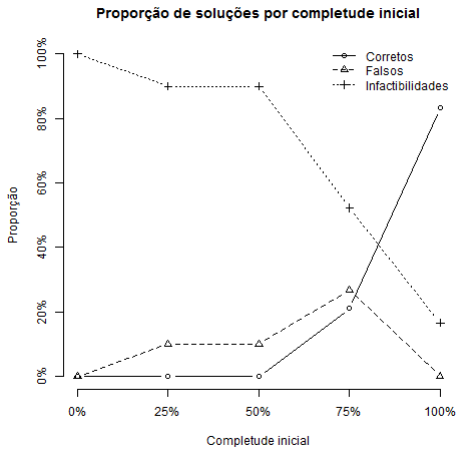

Figura 1. Proporções de soluções obtidas por completude inicial.

V. Considerações Finais

Os resultados preliminares mostram um bom desempenho no processamento de modelos com completude superior a 50%, apesar de que, para valores inferiores, tenha se mostrado pouco efetivo. Trabalhos futuros envolvem a pesquisa de uma abordagem suplementar para lidar com esses casos, além do desenvolvimento de métricas associadas à qualidade dos modelos.

Referências

[1] S. Kambhampati, *Model-lite planning for the web age masses: the challenges of planning with incomplete and evolving domain models* in Proceeding of AAAI Conference on Artificial Intelligence 22, pp. 1601-1604, 2007.

[2] P. Bertoli, M. Pistore and P. Traverso, *Automated composition of web services via planning in asynchronous domains* Artificial Intelligence Journal, vol. 174, pp. 316-361, 2010.

[3] J. Blythe, E. Deelman and Y, Gil, *Automatically composed workflows for grid environments* IEEE Intelligent Systems, vol. 19, pp. 16-23, 2004.

[4] J. Hoffmann, P. Bertoli, and M. Pistore, *Web service compositions as planning, revisited: in between background theories and initial state uncertainty* in Proceeding of AAAI Conference on Artificial Intelligence 22, pp. 1013-1018, 2007.

[5] H. H. Zhuo, T.Ñguyen and S. Kambhampati, *Refining incomplete planning domain models through plan traces* in Proceeding of IJCAI 23, pp. 2451-2457, 2013.

[6] H. Kautz, B.Selman and J. Hoffmann, *Satplan: planning as satisfiability* in Abstracts of International Planning Competition 5, 2006.