Usando Redes de Petri e Resolvedores ISCAS para Tratar Planejamento como Satisfatibilidade

Razer Montaño[1], **Marcos Castilho**[1], **Fabiano Silva**[1], **Luis Künzle**[1]

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Centro Politécnico – Jardim das Américas
Caixa Postal 19.081 – 81.531-980 – Curitiba – PR – Brazil

{razer, fabiano, marcos, kunzle}@inf.ufpr.br

Abstract. *This work presents an approach to solve classical planning problems in AI as a non-clausal satisfiability problem. An SAT instance in ISCAS format is generated by translating planning problem based in a Petri Net, in which submarking-reachability problem must be solved. So, the SAT instance is solved by well-known non-clausal SAT solvers.*

Resumo. *Este trabalho apresenta uma abordagem para resolver o problema de planejamento clássico em IA como um problema de satisfatibilidade não clausal. Uma instância SAT em formato ISCAS é gerada pela conversão do problema de planejamento tendo como base uma rede de Petri na qual o problema de alcançabilidade de sub-marcação deve ser resolvido. A instância ISCAS é então resolvida usando-se resolvedores SAT não-clausais conhecidos.*

1. Introdução

Planejamento é uma subárea do Raciocínio sobre Ações. Um problema de planejamento clássico é dado por um estado inicial, um estado objetivo e um conjunto de possíveis ações a serem tomadas. Uma solução para esse problema é uma sequência de ações que, ao serem executadas, transformam o estado inicial no estado objetivo. Os programas que resolvem esses problemas e encontram esse conjunto de ações são os planejadores.

Em [Silva et al. 2000] foi mostrado o relacionamento entre planejamento e alcançabilidade em Redes de Petri (RdP). O algoritmo PETRIPLAN possibilita que um grafo de planos possa ser transformado em uma RdP ordinária e limitada [Murata 1989]. A vantagem no uso de RdP é que a própria dinâmica da rede, sem o uso de estruturas auxiliares, representa pré-condições e efeitos da execução de uma ação, bem como conflitos (também conhecidos como *mutex*). Entretanto, alcançabilidade em RdP é um processo exponencial [Esparza 1996], para redes que possuem conflitos. Já para redes acíclicas e livres de conflitos, este processo é polinomial.

Nesta abordagem são usadas as RdP pelas vantagens estruturais desta representação. Os conflitos são representados na própria estrutura, a dinâmica da rede já representa efeitos e pré-condições de ações, há a possibilidade de se tratar problemas temporais, entre outros.

Existem muitas abordagens para o tratamento do problema de planejamento. O SATPLAN [Kautz and Selman 1992] trata planejamento como um problema de satisfatibilidade. O GRAPHPLAN [Blum and Furst 1995] usa um grafo construído a partir de

uma descrição STRIPS [Fikes and Nilsson 1971] para diminuir a representação do espaço de busca do problema, conhecido como *Grafo de Planos*.

O HSP [Bonet and Geffner 2001] é um planejador baseado em funções heurísticas que guiam um processo de subida de encosta. Usando-se esta mesma estratégia, tem-se o FF [Hoffmann and Nebel 2001] e o LPG [Gerevini et al. 2003].

Em 1985, Drummond [Drummond 1985] sugere pela primeira vez o uso de RdP no tratamento de problemas de planejamento. Em 2000 surge o PETRIPLAN [Silva et al. 2000], que trata planejamento como alcançabilidade. Alcançabilidade em RdP também é tratada através da técnica de desdobramento[1] [Hickmott et al. 2007].

O objetivo deste trabalho é mostrar uma nova técnica para se resolver o problema do planejamento representado como uma RdP usando SAT. Um problema em PDDL [Mcdermott et al. 1998] é lido e representado como uma RdP acíclica e esta rede é expandida e transformada em uma instância SAT. Em [Montaño et al. 2007], esta instância era uma fórmula proposicional na NNF (*Negation Normal Form*), a qual era transformada para o formado FNNF (*Factored Negation Normal Form*), através de um tableaux KE. Esta fórmula resultante possui propriedades que permitem se efetuar a verificação SAT em tempo polinomial.

O principal ponto que gera problemas de desempenho nesta técnica é a transformação de NNF para FNNF. Neste trabalho, para evitar esta transformação, a instância SAT é gerada em formato ISCAS e resolvida diretamente pelo NFL-SAT [Jain and Clarke 2009], um resolvedor SAT escrito para verificação de circuitos eletrônicos de processadores.

Planejamento e alcançabilidade em RdP são problemas equivalentes [Silva et al. 2000]. RdP possui uma dinâmica que é aproveitada para enriquecer a representação do problema, evitando estruturas auxiliares para representar e propagar conflitos. Esta é a principal diferença nesta abordagem com o BLACKBOX [Kautz and Selman 1999] e SATPLAN [Kautz and Selman 1992], que precisam de passos de propagação de conflitos. Além disso, RdP fornecem um mesmo formalimo para o tratamento de planejamento clássico e planejamento com recursos ou informações temporais.

A abordagem não-clausal para se resolver SAT é ideal neste trabalho, pois a RdP resultante do problema de planejamento está naturalmente na NNF. Uma transformação para CNF (formato comum de entrada dos resolvedores SAT) acrescentaria mais um passo computacionalmente caro de transformação e poderia destruir a estrutura original da fórmula, informação que pode ser utilizada para otimizar o processo.

Na seção 2 descreve-se brevemente a teoria de redes de Petri. Na seção 3 trata-se dos métodos para SAT não-clausal, enquanto que na seção 4 estes resultados são aplicados ao problema de alcançabilidade em RdP, cuja solução nos dá, pela equivalência dos problemas, a solução do problema de planejamento original.

[1] Do inglês *unfolding*

2. Redes de Petri

Rede de Petri (RdP) é uma ferramenta de modelagem com uma representação gráfica e com uma sólida base matemática de análise de propriedades estruturais e comportamentais dos modelos nela construídos. Seu campo de aplicação são os sistemas dinâmicos a eventos discretos, ou seja, que possuem características de paralelismo, concorrência, não determinismo e assincronia [Murata 1989].

Graficamente, uma rede de Petri é representada por um grafo direcionado bipartido constituído por lugares e transições representados, respectivamente, por círculos e retângulos. Arcos, representados por setas, associam lugares a transições e transições a lugares (pré e pós-condições da ocorrência de eventos), enquanto que marcas (caracterizadas por um ponto preto) representam o estado atual da rede.

Definição 1 (Redes de Petri) *Uma rede de Petri* $\mathtt{RdP}$ *é uma tupla* $\langle \mathtt{P}, \mathtt{T}, \mathtt{F}, \mathtt{M_0} \rangle$ *onde* $\mathtt{P}$ *é um conjunto finito não-vazio de lugares,* $\mathtt{T}$ *é um conjunto finito não-vazio de transições,* $\mathtt{F}$ *é uma dupla* $\langle \mathtt{Pre} : (\mathtt{P} \times \mathtt{T}) \rightarrow \mathbb{N}, \mathtt{Pos} : (\mathtt{P} \times \mathtt{T}) \rightarrow \mathbb{N} \rangle$ *que representa a incidência dos arcos que ligam lugares às transições e transições à lugares, respectivamente, e* $\mathtt{M_0} : \mathtt{P} \rightarrow \mathbb{N}$ *é a marcação inicial da rede.*

O comportamento dinâmico da rede é dado pelo disparo das transições, mas para isso é necessário que a transição esteja habilitada. A condição de habilitação de uma transição $\mathtt{t}$ é dada por $\forall \mathtt{p} \in \mathtt{P}, \mathtt{M(p)} \geq \mathtt{Pre(p,t)}$, onde $\mathtt{M(p)}$ define o número de marcas em um lugar $\mathtt{p} \in \mathtt{P}$. Com a transição habilitada, então o disparo pode ocorrer. Assim, os efeitos do disparo da transição $\mathtt{t}$ é o consumo de $\mathtt{Pre(p,t)}$ marcas de cada lugar pertencente ao seu pré-conjunto e colocar $\mathtt{Pos(p,t)}$ marcas em cada lugar do seu pós-conjunto, levando a rede a uma nova marcação, ou seja, $\mathtt{M'} = \mathtt{M} - \mathtt{Pre(p,t)} + \mathtt{Pos(p,t)}$.

Quando uma sequência de transições $\mathtt{s} = \{\mathtt{t_1}, \mathtt{t_2}, ..., \mathtt{t_n}\}$ é disparada levando a rede de uma marcação $\mathtt{M_0}$ a uma nova marcação $\mathtt{M'}$, então é dito que $\mathtt{M'}$ é alcançável a partir de $\mathtt{M_0}$. O conjunto de marcações alcançáveis ($\mathtt{RM}$) são todos os estados que uma dada rede de Petri pode alcançar a partir do disparo de alguma sequência de transições.

Definição 2 (Problema de Alcançabilidade) *Um problema de alcançabilidade consiste em verificar se existe uma sequência de disparos de transições tal que a partir da marcação inicial* $\mathtt{M_0}$ *se possa alcançar uma dada marcação* $\mathtt{M}$*.*

Cada lugar da rede também pode ter uma capacidade máxima $\mathtt{k}$ a ele associada. Isto significa que o número de marcas do lugar não poderá exceder sua capacidade e que isto deve ser respeitado para todas as marcações acessíveis da rede.

Definição 3 (Rede de Petri k-limitada) *Uma rede de Petri* $\mathtt{RdP} = (\mathtt{P}, \mathtt{T}, \mathtt{F}, \mathtt{M_0})$ *é dita k-limitada se, e somente se, para todas as marcações alcançáveis de* $\mathtt{RM}$*,* $\forall \mathtt{p} \in \mathtt{P}, \mathtt{M(p)} \leq \mathtt{k(p)}$*, onde* $\mathtt{k} \in \mathbb{N}$ *é o número máximo de marcações de um lugar. As redes 1-limitadas também são chamadas de redes seguras.*

O PETRIPLAN [Silva et al. 2000] é o algoritmo de planejamento que recebe como entrada um problema de planejamento em PDDL e gera uma RdP Lugar/Transição, acíclica, onde cada transição pode disparar uma única vez. A geração é similar à geração do Grafo de Planos [Blum and Furst 1995], exceto pelo fato de que a própria dinâmica da rede já representa os conflitos e suas propagações. Resolver alcançabilidade nesta rede é o mesmo que resolver o problema de planejamento subjacente.

O estado inicial é representado por marcações efetuadas na rede. O estado objetivo são lugares onde as marcas devem estar após o disparo de transições. Os conflitos são lugares com uma marca, que pode ser consumida por, neste caso, no máximo duas transições. Portanto, o objetivo é encontrar uma sequência de disparos na qual a marcação resultante contenha o estado objetivo.

O principal problema de uma rede como esta são os conflitos. Se nesta rede não houvessem conflitos, um simples caminhamento pelos lugares e transições resolveria alcançabilidade. Portanto, o objetivo principal nesta rede é a eliminação dos conflitos. Quando uma sequência consistente de disparos é encontrada, é convertida para a representação de um plano (sequência de ações), resolvendo assim o problema de planejamento.

3. SAT Não-Clausal

SAT é o problema de encontrar uma valoração para uma determinada fórmula lógica que a torne $Verdadeira$. A grande maioria das aplicações usa fórmulas em CNF, em virtude de otimizações que podem ser efetuadas no tratamento desse tipo de fórmula. Entretanto, muitos problemas, quando analisados como satisfatibilidade, geram fórmulas que não estão em CNF, sendo necessária uma conversão prévia para que se trate o problema como SAT tradicional.

Converter uma fórmula qualquer para CNF se faz de duas formas: pela aplicação sucessiva de operações distributivas ou pela transformação de Tseitin [Tseitin 1968].

A aplicação de operações de distribuição tende a aumentar o tamanho da fórmula de forma exponencial, tornando inviável sua aplicação.

A transformação de Tseitin baseia-se em converter uma igualdade, entre a nova variável e um trecho da fórmula, em uma dupla implicação que, de forma direta, é transformada para seu equivalente na CNF. A transformação é feita em tempo linear em relação ao tamanho da fórmula. O problema deste método é que a estrutura original da fórmula, que poderia ser aproveitada pelo resolvedor, é perdida.

Assim, o processo de conversão para CNF e aplicação de um resolvedor SAT para CNF ou eleva drasticamente o tempo de execução e consumo de memória do planejador, ou perde informações estruturais da fórmula original, tornando interessante o estudo de resolvedores SAT para fórmulas não-clausais [Thiffault et al. 2004].

Uma abordagem para resolver esse problema foi proposta por Montaño [Montaño et al. 2007], na qual a RdP era transformada para uma fórmula proposicional, baseada nos conflitos da rede. Esta fórmula estava na NNF e era convertida para d-DNNF [Darwiche and Marquis 2002], usando-se um tableau KE. Qualquer modelo desta fórmula era um modelo para o problema de planejamento subjacente.

O problema desta abordagem é que, apesar da busca por um modelo de uma fórmula na d-DNNF ser polinomial, transformar uma fórmula da NNF para d-DNNF é P-SPACE [R. Hähnle and N. Murray and E. Rosenthal 2005]. Assim sendo, o melhor caminho a ser tomado é evitar esta transformação, usando-se resolvedores SAT diretamente em fórmulas na NNF.

O formato ISCAS é comumente usado para representar circuitos de processado-

res para validação. As fórmulas lógicas usadas normalmente contém muitas variáveis e muitos operadores, e qualquer conversão para d-DNNF ou CNF seria inviável, em termos de recursos computacionais, e poderia desfazer a estrutura inicial da fórmula [Thiffault et al. 2004].

O NFLSAT [Jain and Clarke 2009] é um resolvedor SAT para fórmulas não-clausais, usando uma variação do algoritmo DPLL. Este resolvedor transforma a fórmula para uma estrutura em árvore, com as negações somente nas folhas (NNF). Esta representação interna favorece a aplicação de modernas técnicas como: observação de literais, retrocesso não-cronológico e aprendizado de cláusulas.

4. Alcançabilidade em Redes de Petri via SAT para Formas Não-Clausais

O algoritmo PETRIPLAN [Silva et al. 2000] gera uma rede de Petri Lugar/Transição acíclica, muito similar ao grafo de planos. Nesta rede, cada transição pode disparar no máximo uma vez e todos os lugares possuem no máximo dois arcos de saída. A situação inicial é dada por marcas em lugares que representam os estados iniciais e nos conflitos. A situação objetivo são lugares onde estas marcas deverão estar após a execução das ações. A execução de ações é dada pelo disparo de transições e escolha de conflitos.

A principal diferença entre o grafo de planos e a rede de Petri está no fato dos conflitos serem representados como meta-informações no grafo, enquanto que na rede de Petri eles são parte da rede. Ainda, o grafo de planos é uma estrutura estática e os métodos de busca devem percorrer a estrutura até encontrar um caminho livre de conflitos. Na rede de Petri existe uma dinâmica associada, fato que permite reduzir sensivelmente o número de conflitos na rede em relação ao grafo de planos.

A rede obtida pelo PETRIPLAN representa o comportamento dinâmico da aplicação de ações, partindo do estado inicial em busca de um estado final. Resolver planejamento usando esta RdP é encontrar todas as transições a serem disparadas, na ordem correta, para que o estado objetivo seja encontrado a partir do estado inicial. Esta sequência de ações é chamada *plano*.

Na busca pelo estado final encontram-se os conflitos. Em uma rede sem conflitos, o estado final é obtido, se for possível obtê-lo, de forma direta pelo caminhamento nesta rede. A presença de conflitos indica que existem transições que não podem ser disparadas em conjunto para a obtenção do mesmo plano, portanto são mutamente exclusivas.

Isto posto, se os conflitos da RdP forem eliminados de forma consistente, um simples caminhamento da rede é necessário para a obtenção de um plano, se este existir. Portanto, o objetivo aqui é eliminar esses conflitos.

O procedimento proposto pode ser visualizado através do Algoritmo 1.

4.1. Método original

A abordagem original deste trabalho [Montaño et al. 2007] baseia-se no fato de que o problema de satisfatibilidade em uma fórmula na *Deterministic Decomposable Negation Normal Form* (d-DNNF) é polinomial [Darwiche and Marquis 2002]. Também são polinomiais a contagem e enumeração de modelos. Assim sendo, foi proposto um pro-

Algoritmo 1 Planejador

```
Carrega o PDDL
repeat
  Expandir a rede até que os objetivos sejam encontrados consistentemente
  Gerar a fórmula no formato ISCAS
  Aplicar resolvedor NFLSAT
  if NFLSAT encontrou valoração para a fórmula then
    Constrói o plano a partir da valoração
  end if
until encontrar plano
```

cesso de conversão da fórmula lógica obtida da RdP para d-DNNF, baseado em Tableau KE.

Fórmulas na d-DNNF (decomponíveis e determinísticas) são um caso particular de fórmulas na *Negation Normal Form* (NNF). Uma sentença está na NNF se é construída a partir de literais, usando-se somente disjunções e conjunções.

Definição 4 (Decomponibilidade) *Dada uma fórmula na NNF, para cada conjunção $\alpha_1 \wedge \ldots \wedge \alpha_n$, se uma variável não aparece em mais de um termo α_i, isto é, os átomos encontrados em um termo desta conjunção não são os mesmos encontrados em outros termos, então ela é decomponível.*

Uma fórmula decomponível permite que certos tipos de problemas computacionais sejam decompostos em problemas menores, dado que suas subsentenças não compartilham átomos. Sempre que um nodo and for encontrado, cada um de seus ramos terá, seguramente, informações diferentes, podendo ser computado de forma independente.

Definição 5 (Determinismo) *Dado uma fórmula na NNF, se para toda disjunção $\vee_i \alpha_i$, todo par de termos α_i e α_j, onde $i \neq j$, $\alpha_i \wedge \alpha_j$ é uma contradição, isto é, são mutuamente exclusivos, então a fórmula é determinística.*

Decomponibilidade é a propriedade que torna o teste de satisfatibilidade computacionalmente tratável. É a propriedade de determinismo que torna a contagem de modelos e sua enumeração tratável.

Para se transformar uma teoria proposicional em d-DNNF deve-se aplicar, recursivamente, uma operação chamada *condicionamento*[2] [Palacios et al. 2005]. Esta operação é derivada da identidade conhecida como expansão de Shannon dada por:

$$\mathcal{F} = (\mathcal{F}[V/p] \wedge p) \vee (\mathcal{F}[F/\neg p] \wedge \neg p)$$

onde $\mathcal{F}[x/p]$ indica a troca de todas as ocorrências de p em $\mathcal{F}$ por x.

Uma fórmula na *Factored Negation Normal Form* (FNNF) também está na d-DNNF, mas regras de identidade, simplificações, são aplicadas à fórmula para geração de fórmulas menores e equivalentes.

[2]Do inglês *conditioning*

O tableau KE [D'Agostino and Mondadori 1994] é um procedimento de prova baseado em tableau, mas que inclui a regra de *cut*, que classicamente não é considerada. Conforme mostrado em [R. Hähnle and N. Murray and E. Rosenthal 2005] o procedimento de Tableau KE, pode ser usado para a geração da FNNF.

No procedimento de transformação para FNNF somente são usadas a regra *cut*, regras α e regras de transformação de tableau. As regras β são substituídas por um mecanismo de propagação de restrições, conhecido como *simplificação* [Massacci 1998]. Este mecanismo tem o mesmo propósito que a subjugação para a resolução e que a regra de cláusulas unitárias para o procedimento de Davis-Putnam.

O tableau resultante tem literais em todos os nodos ou a árvore inteira será reduzida a $Falso$ ou $Verdadeiro$. Esse tableau é chamado *Tableau FNNF saturado*.

Aqui, não há necessidade de se obter um tableau saturado pois qualquer ramo saturado desse tableau já é um modelo para a fórmula que está sendo processada. Este processo de parada caracteriza um o procedimento como um SAT não-clausal, sendo que do ramo saturado do Tableau KE se consegue extrair um modelo para a fórmula em NNF original. Esta valoração é transformada em cortes na RdP original, sendo que esses cortes incidem sobre os conflitos, já que a fórmula é formada por informações provenientes destes. Esses cortes resolvem todos os conflitos da rede e, a partir deste ponto, pode-se aplicar um procedimentos de busca para encontrar o plano de forma eficiente, pois não será efetuado nenhum *backtracking*.

Apesar de SAT ser polinomial em uma fórmula na FNNF, a conversão de NNF para FNNF é P-SPACE [R. Hähnle and N. Murray and E. Rosenthal 2005], o que degrada todo o processo. Isto posto, deve-se obter um processo que não exiga esta transformação e no qual se possa fazer SAT em uma fórmula não-clausal de maneira eficiente.

Analisando-se o formato não-clausal ISCAS, percebeu-se que a RdP poderia ser facilmente convertida para esta representação. Ademais, as fórmulas geradas seriam menores, pois o uso e reaproveitamento de operadores neste formato pode ser usado para representar lugares ou transições que são visitados várias vezes, uma vez só.

Assim, o processo apresentado aqui, propõem que a RdP seja transformada diretamente para o formato ISCAS e que neste resultado um resolvedor seja usado. Diferentemente da proposta original, o passo de transformação para FNNF pode ser removido, já que existem resolvedores ISCAS eficientes (como o NFLSAT).

4.2. Geração de fórmulas ISCAS a partir de uma Rede de Petri

A partir da RdP original, partindo de cada lugar pertencente ao estado objetivo, varre-se a rede até os lugares iniciais construindo uma fórmula lógica onde seus átomos representam unicamente os conflitos. A retirada dos conflitos da rede se dá a partir de um resolvedor SAT para fórmulas não-clausais, que dará valores aos literais dessa fórmula, neste caso o NFLSAT. Na estrutura da RdP, esta valoração indica a eliminação dos conflitos através da imposição de uma escolha.

Nas RdP geradas pelo PETRIPLAN usado nas implementações, os conflitos são sempre binários (um lugar possuindo no máximo dois arcos de saída), portanto pode-se usar uma variável proposicional para representar cada conflito, e estas são as entradas (INPUT) da fórmula. Por exemplo, na Figura 1 $\neg p$ indica o disparo da transição t_1 e p

indica o disparo da transição t_2.

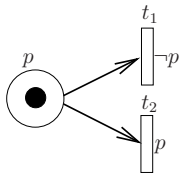

Figura 1. Conflito com a proposição p associada

Analisando-se a estrutura da RdP, percebe-se que os lugares pertencentes ao estado final podem ser descritos como uma fórmula baseada nas variáveis proposicionais associadas aos conflitos. Caso cada um dos lugares que chegam a t tenham fórmulas associadas ($\Delta_1, ..., \Delta_n$), como mostra a Figura 2, a fórmula associada a t é $AND(\Delta_1, ..., \Delta_n)$. Caso cada uma das transições que chegam a p tenham fórmulas associadas ($\Delta_1, ..., \Delta_n$), como mostra a Figura 3, a fórmula associada a p é $OR(\Delta_1, ..., \Delta_n)$.

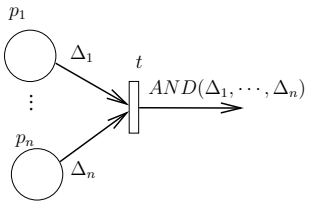

Figura 2. Conjunção na RdP com fórmula associada

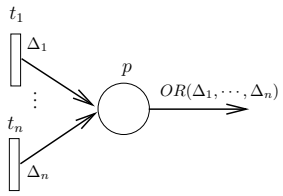

Figura 3. Disjunção na RdP com fórmula associada

Seja M_o a marcação final e M_i a marcação inicial da RdP. Seja $\theta(V)$ a quantidade de elementos de V, onde V é um vetor, diferentes de zero e $\mu(t)$, definido sobre uma transição t, indicando se t é uma transição de um conflito, isto é, $\mu(t)$ é V se $\exists p,\ Pre(p,t) > 0\ \wedge\ \theta(Pre(p,.)) > 1$ e F caso contrário. Assume-se que existe uma ordenação qualquer entre as transições, que pode ser a ordem de inserção na fórmula.

A fórmula $\mathcal{F}$ que representa os conflitos de uma RdP é dada por:

$$\mathcal{F} \quad = \bigwedge_i L(p_i),\ p_i \in M_o$$

$$L(p) = \begin{cases} \bigvee_i T(t_i);\ \forall t_i,\ Pos(p,\ t_i)\ >\ 0 & se\ \theta(Pos(p,\ .))\ \geq\ 1 \\ V & se\ \theta(Pos(p,\ .))\ =\ 0\ e\ p\ \in\ M_i \\ F & se\ \theta(Pos(p,\ .))\ =\ 0\ e\ p\ \notin\ M_i \end{cases}$$

$$T(t) = \begin{cases} \bigwedge_i \alpha(p_i,\ t);\ \forall p_i,\ Pre(p_i,\ t)\ >\ 0 & se\ \theta(Pre(.,\ t))\ \geq\ 1 \\ F & se\ \theta(Pre(.,\ t))\ =\ 0 \end{cases}$$

$$\alpha(p,\ t) = \begin{cases} L(p) & se\ \neg\mu(t) \\ \beta(p,t)\ \wedge L(p) & se\ \mu(t) \end{cases}$$

$$\beta(p,\ t) = \begin{cases} p & se\ \exists t', Pre(p,\ t')\ >\ 0\ e\ t\ >\ t' \\ \neg p & se\ \exists t',\ Pre(p,\ t')\ >\ 0\ e\ t\ <\ t' \end{cases}$$

Baseado na estrutura de geração da fórmula, os únicos conectivos presentes são conjunções e disjunções. Ademais, as negações possuem escopo limitado às variáveis ligadas aos conflitos, portanto a fórmula gerada está na NNF.

Como esta fórmula possui somente literais atrelados a conflitos, e é gerada a partir da interpretação dos lugares e transições, qualquer modelo (valoração que a satisfaça) pode ser interpretado como uma configuração consistente dos conflitos da rede original, eliminando-os, permitindo assim disparos em transições para a obtenção do estado final. Isto indica que existe um plano para o problema de planejamento subjacente. No caso da fórmula ser insatisfatível, conclui-se que não há configuração consistente para os conflitos, de forma que se chegue ao estado final, indicando que não existe plano para o problema de planejamento associado.

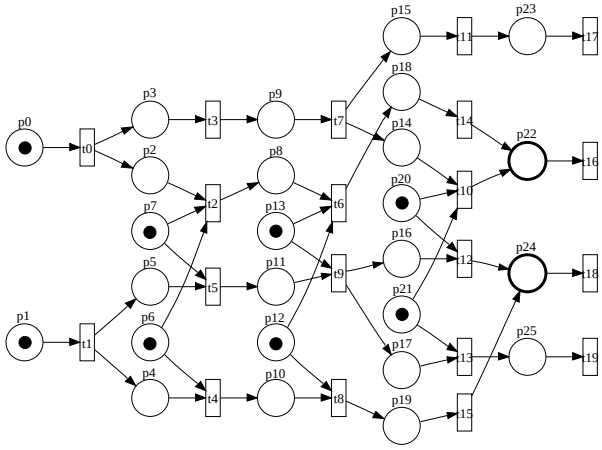

Figura 4. Problema completo com marcações

Como exemplo, a Figura 4 mostra uma RdP com uma marcação inicial (círculos pretos nos lugares) e com uma marcação final (lugares com contornos mais fortes). Nesta rede, deve-se extrair uma fórmula lógica para $p22 \wedge p24$, através de um caminhamento reverso partindo dos objetivos até o estado inicial. A fórmula obtida é:

```
INPUT(p6)
INPUT(p7)
INPUT(p12)
INPUT(p13)
INPUT(p20)
INPUT(p21)
OUTPUT(out)
not_p6 = NOT(p6)
not_p7 = NOT(p7)
not_p12 = NOT(p12)
not_p13 = NOT(p13)
not_p20 = NOT(p20)
not_p21 = NOT(p21)
aux1 = AND(not_p7, not_p6,
           not_p12, not_p13)
aux2 = AND(not_p20, not_p21)
p22 = OR(aux1, aux2)
aux3 = AND(p6, p12)
aux4 = AND(p7, p13, p20)
p24 = OR(aux3, aux4)
out = AND(p22, p24)
```

Aplicando-se o NFLSAT, pode-se obter um modelo, como por exemplo: $\{p6, p7, p12, \neg p20, \neg p21\}$. Como este modelo representa escolhas de conflito, a RdP pode ser cortada, eliminando-se o que não faz parte da escolha. Obtém-se então a rede da Figura 5, na qual um caminhamento simples retorna o plano do problema de planejamento subjacente.

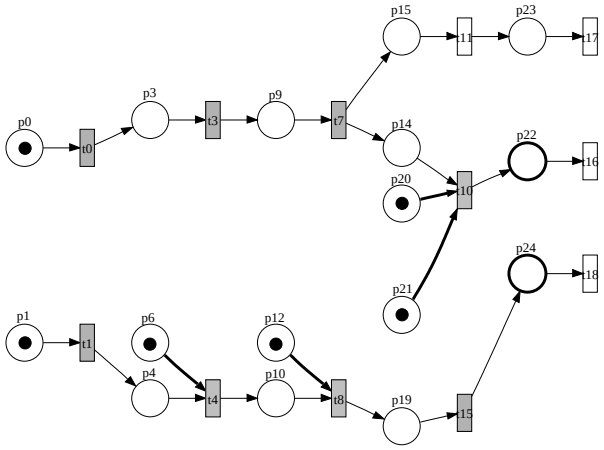

Figura 5. Rede de Petri cortada

5. Experimentos

A implementação efetuada baseia-se no algoritmo PETRIPLAN [Silva et al. 2000] para leitura e representação de um problema de planejamento em PDDL. Com a RdP carregada, o mesmo processo de geração de fórmulas mostrado em Montaño [Montaño et al. 2007] é efetuado, mas aqui o formato gerado é ISCAS, conforme mostrado na seção anterior.

Com a fórmula obtida, o resolvedor NFLSAT é chamado para resolver os conflitos. Se uma solução for obtida, ela é transformada em cortes na rede para que um processo de caminhamento, logo após, determine o plano. Caso uma solução não seja obtida, então uma nova expansão na rede é feita e o processo todo se inicia.

A Tabela 1 mostra os tempos obtidos para a resolução de alguns problemas usando-se a abordagem original de Montaño, SATPLAN e a nova técnica (RDP_ISCAS). Conforme pode-se observar, a abordagem original na qual a fórmula é convertida para d-DNNF não é viável, visto que quase todos os problemas geram estouro no uso de recursos da máquina.

Tabela 1. Comparação de Tempos SATPLAN e RDP_ISCAS (em segundos). MEM indica que o procedimento esgotou recursos de memória antes de terminar. * indica que o problema experimentado não era tipado.

Problema	Expansões	Original	SATPLAN	RDP_ISCAS
Sussman	3	0.04	0	0.01
Blocks-04	3	MEM	0.01	0.088
Blocks-05	6	MEM	0.02	0.317
Blocks-06	6	MEM	0.03	0.97
Blocks-10-2	15	MEM	0.37	78.149
Gripper 2g 2b	3	0.05	0.01	0.01
Gripper 2g 4b	7	MEM	0.04	0.031
Gripper 2g 6b	11	MEM	0.43	5.57
Gripper 2g 8b	15	MEM	19.09	130.95
Logistics 1	9	MEM	0.04*	34.709

Os resultados são preliminares mas promissores, visto que o SATPLAN usa uma

das mais eficientes técnicas para planejamento[3]. Apesar de, na média, os tempos serem maiores do que o SATPLAN, percebe-se uma aproximação de resultados, indicando que a implementação de técnicas de otimização podem levar a resultados melhores.

Para melhorar o resolvedor SAT pode-se implementar novas técnicas e considerar informações estruturais da fórmulas ISCAS, visto que há um padrão de geração da fórmula. Para melhorar a codificação do problema, deve-se usar um outro substrato teórico que, ou que diminua o número de operadores da fórmula, ou produza fórmulas com uma estrutura mais alinhada com as técnicas do resolvedor SAT.

6. Considerações

Ao contrário do BLACKBOX, que usa um grafo de planos como representação interna do problema de planejamento, aqui é usado uma rede de Petri. Como vantagens tem-se: a dispensa no uso de estruturas auxiliares para representar conflitos e a estrutura da rede servindo como mecanismo de propagação de ações, pré-condições e efeitos.

Percebe-se com este trabalho que a técnica de representação de problemas de planejamento como redes de Petri e seu posterior processamento com resolvedores SAT para NNF está indo ao encontro dos resultados obtidos pelo SATPLAN, com a vantagem de se ter um modelo que naturalmente representa o problema.

Os resultados são preliminares, mas dois trabalhos estão sendo desenvolvidos em paralelo, que terão implicação direta nesses resultados: melhoramento no resolvedor NFLSAT e melhoramento no processo de geração das fórmulas ISCAS, usando o novo algoritmo PETRIPLAN, a Rede de Planos [Silva 2005].

Referências

Blum, A. and Furst, M. (1995). Fast planning through planning graph analysis. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95)*, pages 1636–1642.

Bonet, B. and Geffner, H. (2001). Heuristic search planner 2.0. *AI Magazine*, 22(3):77–80.

D'Agostino, M. and Mondadori, M. (1994). The taming of the cut: Classical refutations with analytic cut. *Journal of Logic and Computation*, 4(3):285–319.

Darwiche, A. and Marquis, P. (2002). A Knowledge Compilation Map. *Journal of Artificial Intelligence Research*, 17:229–264.

Drummond, M. E. (1985). Refining and extending the procedural net. In *Proceedings of the 9th international joint conference on Artificial intelligence - Volume 2*, pages 1010–1011, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Esparza, J. (1996). Decidability and complexity of petri net problems - an introduction. In *Petri Nets*, pages 374–428.

Fikes, R. and Nilsson, N. (1971). Strips: A new approach to the application of theorem proving to problem solving. Technical Report 43r, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025. SRI Project 8259.

[3]SATPLAN venceu a competição de planejadores na quarta edição do IPC (*4th International Planning Competition*) e empatou em primeiro lugar em sua quinta edição

Gerevini, A., Saetti, A., and Serina, I. (2003). Planning through stochastic local search and temporal action graphs in lpg. *J. Artif. Int. Res.*, 20:239–290.

Hickmott, S., Rintanen, J., Thiébaux, S., and White, L. (2007). Planning via petri net unfolding. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1904–1911, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Hoffmann, J. and Nebel, B. (2001). The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302.

Jain, H. and Clarke, E. M. (2009). Efficient SAT solving for Non-Clausal formulas using DPLL, graphs, and watched cuts. In *46th Design Automation Conference (DAC)*.

Kautz, H. and Selman, B. (1999). Unifying SAT-based and graph-based planning. In Minker, J., editor, *Workshop on Logic-Based Artificial Intelligence, Washington, DC, June 14–16, 1999*, College Park, Maryland. Computer Science Department, University of Maryland.

Kautz, H. A. and Selman, B. (1992). Planning as satisfiability. In *Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI'92)*, pages 359–363.

Massacci, F. (1998). Simplification: A general constraint propagation technique for propositional and modal tableaux. *Lecture Notes in Computer Science*, 1397:217–231.

Mcdermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., and Wilkins, D. (1998). PDDL - the planning domain definition language. Technical report, CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.

Montaño, R., Silva, F., Castilho, M., and Kunzle, L. (2007). Planejamento como satisfatibilidade: uma abordagem não-clausal. In *Proceedings of the VI Encontro Nacional de Inteligência Artificial (ENIA)*.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

Palacios, H., Bonet, B., Darwiche, A., and Geffner, H. (2005). Pruning conformant plans by counting models on compiled d-DNNF representations. In *ICAPS*, pages 141–150.

R. Hähnle and N. Murray and E. Rosenthal (2005). Normal forms for knowledge compilation. *Lecture Notes in Computer Science*, 3488:304–313.

Silva, F. (2005). *Rede de Planos: Uma Proposta para a Solução de Problemas de Planejamento em Inteligência Artificial usando Redes de Petri*. PhD thesis, CEFET, Curitiba PR, Brasil.

Silva, F., Castilho, M., and Künzle, L. (2000). Petriplan: a new algorithm for plan generation (preliminary report). In *Lecture Notes in Artificial Intelligence*, volume 1952, pages 86–95. International Joint Conference IBERAMIA'2000 - SBIA'2000.

Thiffault, C., Bacchus, F., and Walsh, T. (2004). Solving non-clausal formulas with dpll search. In *In SAT*, pages 663–678. Springer.

Tseitin, G. (1968). On the complexity of derivation in propositional calculus. *Studies in constructive mathematics and mathematical logic*, 2(115-125):10–13.