ROADEF'2005 - VISÃO DE PLANEJAMENTO

Silvio do Lago Pereira

8 de outubro de 2003

1 Introdução

O problema proposto pelo ROADEF'2005 [1] consiste em, dados os carros que deverão ser montados num dia, determinar a melhor ordem de montagem tal que certas restrições sejam satisfeitas. Essas restrições podem ser de dois tipos:

- *pintura*: para minimizar a quantidade de solvente gasto na limpeza da máquina (necessária a cada mudança de cor ou a cada n carros pintados, mesmo que não haja mudança de cor), os carros devem aparecer na seqüência de montagem agrupados por cores;
- *montagem*: para não sobrecarregar a linha de produção, carros que demandam operações especiais (*e.g.*, instalação de teto solar ou ar condicionado) devem estar uniformemente distribuídos na seqüência de montagem. Cada operação especial tem associada uma razão da forma N/P, indicando que numa seqüência de P carros, podemos ter no máximo N carros requerendo tal operação. Essas restrições podem ser de *alta* ou *baixa* prioridade.

Enquanto a pintura é uma restrição *rígida*, montagem é uma restrição *frouxa* (*i.e.*, não é possível garantir que todas as restrições de montagem possam ser satisfeitas). Sendo assim, durante a busca da melhor ordem de montagem dos carros, nosso objetivo será atender às restrições de pintura, minimizando o número de violações das restrições de montagem. Ademais, não deverá haver nenhuma compensação entre objetivos, ou seja, uma restrição menos importante não deverá ser satisfeita às custas da violação de uma outra restrição mais importante. Além disso, ao iniciar a busca pela melhor seqüência de montagem para um determinado dia, deveremos levar em conta os últimos carros na seqüência de montagem do dia anterior e, portanto, a contabilização do número total de violações deverá considerar também esses carros (embora a ordem deles não possa mais ser modificada).

2 Descrição do problema

Um problema nesse domínio é descrito através de quatro arquivos: o primeiro deles determina o limite de pinturas, antes que a máquina tenha que ser lavada; o segundo indica a prioridade relativa das restrições consideradas; o terceiro estabelece as razões associadas às operações de montagem; e o último descreve os carros a serem montados. Por exemplo:

1. `paint_batch_limit_.txt`

```
limitation;
10;
```

2. `optimization_objectives.txt`

```
rank;objective name;
1;paint_color_batches;
2;high_priority_level_and_easy_to_satisfy_ratio_constraints;
3;low_priority_level_ratio_constraints;
```

3. `ratios.txt`

```
Ratio;Prio;Ident;
2/4;1;HPRC2;
1/4;1;HPRC3;
...
```

4. `vehicles.txt`

```
Date;SeqRank;Ident;Color;HPRC1;HPRC2;HPRC3;HPRC4;HPRC5;LPRC1;LPRC2;LPRC3;...
2003 26 1;1050;039032521147;2;0;0;0;0;0;0;0;1;0;0;0
2003 27 1;   1;039032440804;3;0;0;0;0;0;0;0;0;0;0;0
2003 27 1;   2;039032620549;5;0;0;0;0;0;0;0;0;0;0;0
```

3 Modelagem em termos de planejamento

Evidentemente, em termos de planejamento, uma solução para o problema proposto deverá ser um plano cuja ordem das ações corresponda à ordem de montagem dos carros. Uma forma natural de conseguir esse mapeamento é associar a cada carro um operador correspondente. Assim, por exemplo, o plano $\langle do913, do127, \ldots \rangle$ indicaria que o carro cujo identificador é 913 seria o primeiro a ser montado, aquele cujo identificador é 127 seria o segundo, *etc.*

Para entender como codificar esses operadores, considere um exemplo simplificado onde precisamos montar um carro com as seguintes características: `Ident = 913, AirCond = false, SunRoof = true`. Além disso, suponha que a razão associada à operação de instalação de teto solar seja 2/5 (como esse carro particular não vai ter ar condicionado, a razão associada a essa última operação é irrelevante) e que a "unidade de duração" corresponda ao período de montagem de um carro. Então, usando PDDL [2], podemos codificar o seguinte operador para o referido carro:

```
(:durative-action do913
  :parameters ()
  :duration   (= ?duration 5)
  :condition  (and (at start (work 913))
                   (at start (>= (sunroof) 1)))
  :effect     (and (at start (not (work 913)))
                   (at start (decrease (sunroof) 1))
                   (at end (increase (sunroof) 1))))
```

Note que os operadores devem ser instâncias específicas para cada um dos carros e não esquemas a serem instanciados pelo planejador. Nesse operador específico, o átomo `(work 913)`, requerido como precondição, é também um efeito negativo do operador. Portanto, fornecendo esse átomo no estado inicial, e garantindo que nenhum operador o produza como efeito, garantimos também que a ação `do913` ocorrerá uma única vez no plano. Além disso, iniciando a variável numérica `sunroof` com o valor 2, e dando à ação a duração 5, conseguimos modelar a razão 2/5 (supondo que as ações possam ser executadas concorrentemente). Observe que uma das condições do operador `do913` é que a variável `sunroof` seja pelo menos 1. Se tal condição não é satisfeita, significa que já existem dois carros com teto solar sendo montados entre os últimos 5 carros da seqüência. Ademais, caso o operador `do913` seja aplicável, o valor da variável `sunroof` é imediatamente decrementado, sendo incrementado novamente apenas no final da duração da ação, após 5 carros terem sido montados.

O diagrama a seguir, onde `x` representa a montagem de um carro com teto solar, ilustra o funcionamento desses operadores. Note que a variável `sunroof` é incrementada como um efeito "retardado" dos operadores.

```
sunroof : 2111001112
sequence:  x--x-----
           ^  ^ ^  ^
           |  | |  |
           +--|-+  |
              +----+
```

Assim, em termos de planejamento, um problema de seqüenciamento de carros nesse domínio poderia ser descrito da seguinte maneira:

```
(define (problem day-1)
  (:domain assembly)
  (:init (work 127)
         (work 913)
          ...
         (= (sunroof) 2)
         (= (aircond) 1)
          ...)
  (:goal (and (not (work 127))
              (not (work 913))
               ...)))
```

4 Conclusão

Conforme mostramos, a modelagem do problema de seqüenciamento de carros como uma tarefa planejamento requer que o planejador seja capaz de considerar a execução concorrente dos operadores, permitindo uma nova ação seja iniciada antes do término de uma ação anterior (e que tenhamos efeitos retardados).

Caso a execução concorrente são seja possível, então a modelagem do problema será muito difícil pois, conforme a seção 3 de [2], PDDL não permite que variáveis númericas sejam usadas em termos da linguagem (*i.e.*, argumentos de predicados ou de ações do domínio). Conseqüentemente, informações sobre o sufixo da seqüência de carros parcialmente construída durante a busca não pode ser facilmente representada e ficará muito difícil resolver o problema usando planejamento.

Referências

[1] NGUYEN, A. *Challenge* ROADEF*'2005: Car Sequencing Problem*, Renault, France, 2003.

[2] FOX, M. & LONG, D. *PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains*, University of Durham, UK, 2003.