

Data Science Academy

www.datascienceacademy.com.br

Introdução à Inteligência Artificial

Tempo, Escalonamento e Recursos



Vimos os conceitos mais básicos, representações e algoritmos para planejamento (sim, o que nós vimos foram os conceitos básicos). Os planejadores empregados no mundo real para planejar e escalonar as operações de espaçonaves, fábricas e campanhas militares são mais complexos; eles estendem a linguagem de representação e também o modo como o planejador interage com o ambiente. Isso permite que os especialistas humanos comuniquem ao planejador o que sabem sobre como resolver o problema. A hierarquia também se presta à construção eficiente do plano porque o planejador pode resolver um problema em nível abstrato, antes de se aprofundar em detalhes.

A representação do planejamento clássico informa o que fazer e em que ordem, mas a representação não pode informar sobre o tempo: quanto tempo uma ação leva e quando ela ocorre. Por exemplo, os planejadores podem produzir um cronograma para uma companhia aérea que informa quais aviões estão designados a quais voos, mas também precisamos saber realmente os horários de partida e de chegada. Esse é o tema do escalonamento. O mundo real também impõe muitas restrições de recursos; por exemplo, uma companhia aérea tem número limitado de pessoal — e o pessoal designado para um voo não pode estar em outro ao mesmo tempo. Existem diversos métodos para representar e resolver problemas de planejamento que incluem restrições temporais e de recursos.

A abordagem adotada é “planejar primeiro, escalonar mais tarde”, isto é, dividimos o problema global em uma fase de planejamento, em que as ações são selecionadas com algumas restrições de ordem para satisfazer aos objetivos do problema, e em uma fase de escalonamento posterior, na qual informações temporais são adicionadas ao plano, a fim de assegurar que ele atenderá às restrições de recursos e de prazos.

Vejamos um problema de escalonamento de linha de produção para montar dois carros, com restrições de recursos. A notação A B significa que a ação A deve preceder a ação B. Essa abordagem é comum em configurações do mundo real de manufatura e logística, em que a fase de planejamento é realizada frequentemente por especialistas humanos. Os métodos automatizados também podem ser utilizados para a fase de planejamento, desde que eles produzam planos apenas com o mínimo de ordenação de restrições necessárias para correção. Os algoritmos GRAPHPLAN, o SATPLAN e os planejadores de ordem parcial podem fazer isso; os métodos baseados em busca produzem planos totalmente ordenados, mas podem ser facilmente convertidos em planos com restrições de ordenação.

Um problema típico de escalonamento de linha de produção, consiste em um conjunto de processos (jobs), cada um dos quais é composto por um conjunto de ações com restrições de ordenação entre elas. Cada ação tem uma duração e um conjunto de restrições de recursos exigidos pela ação. Cada restrição especifica um tipo de recurso (por exemplo, parafusos, chaves, etc...), a quantidade necessária desse recurso e se esse recurso é consumível (por exemplo, os parafusos não estão mais disponíveis para uso) ou reutilizável (por exemplo, um piloto está ocupado durante um voo, mas ficará disponível quando o voo terminar). Os recursos também podem ser produzidos por ações com consumo negativo, incluindo ações de manufatura, crescimento e reabastecimento. Uma solução para um problema de

escalonamento de linha de produção deve especificar os horários de início de cada ação e deve satisfazer todas as restrições de ordenação temporal e as restrições de recursos. Tal como acontece com problemas de busca e planejamento, soluções podem ser avaliadas de acordo com uma função de custo, o que pode ser bastante complicado, com custos de recursos não lineares, custos dependentes do tempo de atraso, e assim por diante. Para simplificar, assumiremos que a função custo é a duração total do plano, que é chamado de makespan. A figura abaixo mostra um exemplo simples: um problema que envolve a montagem de dois carros.

Processos ({AdicionarMotor1 AdicionarRodas1 Inspeccionar1},
{AdicionarMotor2 AdicionarRodas2 Inspeccionar2})

Recursos (GuinchoParaMotor(1), EstaçãoDeRodas(1), Inspetores(2),
PorcasDeRoda(500))

Ação (AdicionarMotor1, DURAÇÃO:30, USO: GuinchoParaMotor(1))

Ação (AdicionarMotor2, DURAÇÃO:60, USO: GuinchoParaMotor(1))

Ação (AdicionarRodas1, DURAÇÃO:30, CONSUMO: PorcasDeRoda(20), USO:
EstaçãoDeRodas(1))

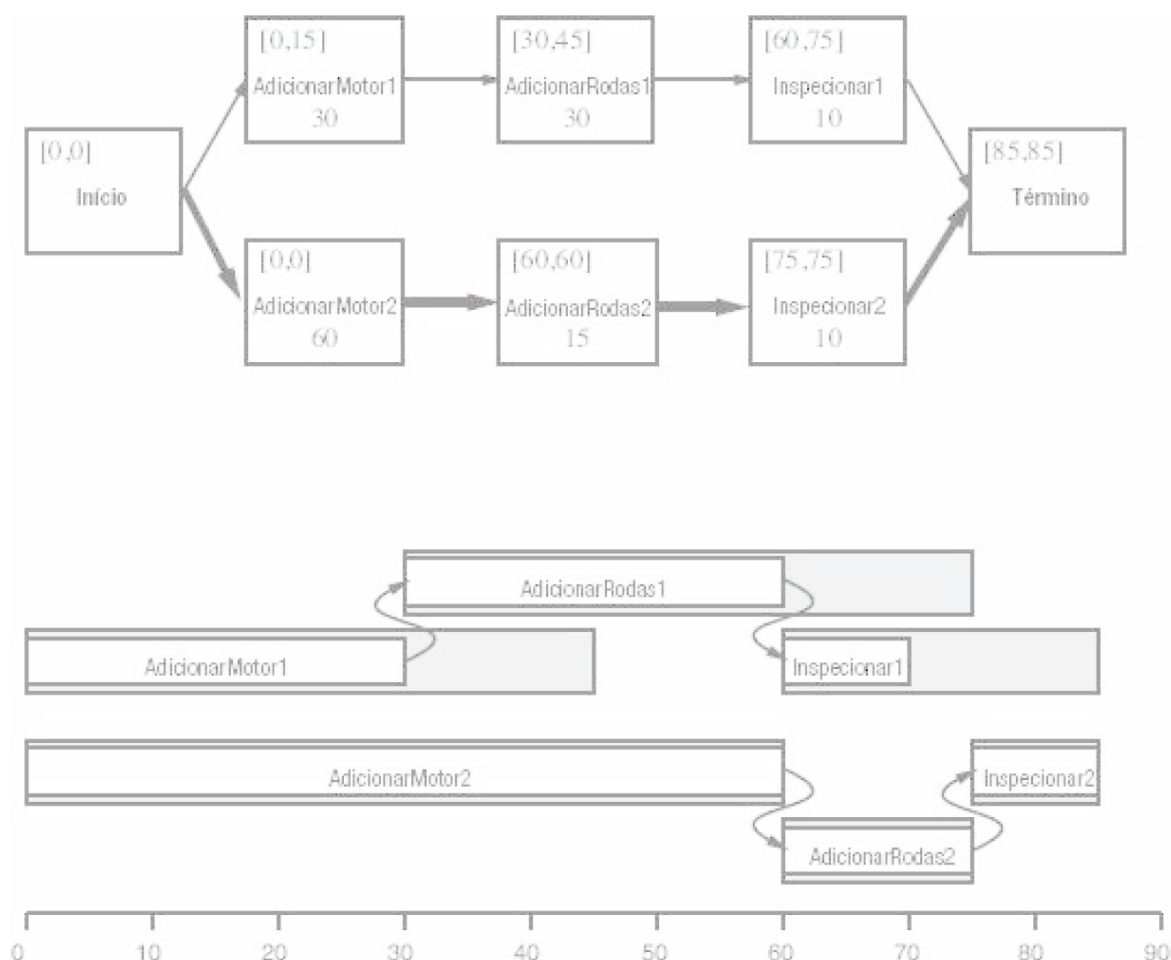
Ação (AdicionarRodas2, DURAÇÃO:30, CONSUMO: PorcasDeRoda(20), USO:
EstaçãoDeRodas(1))

Ação (Inspeccionar, DURAÇÃO:10, USO: Inspetores(1))

O problema consiste em dois processos, cada um da forma [AdicionarMotor, AdicionarRodas, Inspeccionar]. Em seguida, a instrução Recursos declara que existem quatro tipos de recursos e dá a quantidade de cada tipo disponível no início: 1 guincho para levantar o motor, 1 estação de rodas, 2 inspetores e 500 porcas de roda. Os esquemas de ação fornecem a duração e os recursos necessários para cada ação. As porcas de roda são consumidas à medida que as rodas são adicionadas ao carro, enquanto os outros recursos são “emprestados” no início de uma ação e liberados no final da ação. A representação de recursos como quantidades numéricas, como Inspetores(2), em lugar de entidades no meadas, como Inspetor(I1) e Inspetor(I2), é um exemplo de técnica muito geral chamada de agregação. A ideia central da agregação é agrupar objetos individuais em quantidade quando os objetos são todos indistinguíveis no que se refere ao propósito em questão. Em nosso problema de montagem, não importa qual inspetor inspeciona o carro, e, portanto, não há necessidade de fazer a distinção. A agregação é essencial para reduzir a complexidade. Considere o que acontece quando é proposto um escalonamento que tem 10 ações inspeccionar concorrentes, mas há apenas nove inspetores disponíveis. Com os inspetores representados como quantidades, uma

falha é detectada de imediato e o algoritmo realiza o retrocesso para tentar outro escalonamento. Com os inspetores representados como indivíduos, o algoritmo efetua o retrocesso para experimentar todas as 10 maneiras de atribuir inspetores para ações.

Começaremos considerando apenas o problema de escalonamento temporal, ignorando as restrições de recursos. Para minimizar o makespan (duração do plano), deveremos encontrar os tempos de início de todas as ações consistentes com as restrições de ordem fornecidas com o problema. É útil visualizar essas restrições de ordem como um grafo direcionado relativo às ações, como mostrado na figura abaixo. Podemos aplicar o método do caminho crítico (CPM — critical path method) para esse grafo, para determinar o tempo inicial e final possível de cada ação. Um caminho através de um grafo que representa um plano de ordem parcial é uma sequência de ações linearmente ordenadas começando em Início e terminado em Término (por exemplo, existem dois caminhos no plano de ordem parcial neste diagrama abaixo).



A duração de cada ação é dada na parte inferior de cada retângulo. Na solução do problema, calculamos os tempos de início mais cedo e mais tardio como o par [ES, LS], exibido



no canto superior esquerdo. A diferença entre esses dois números é a folga de tempo de uma ação; as ações com zero de folga estão no caminho crítico, como indica as setas em negrito. Parte inferior: a mesma solução mostrada na linha do tempo. Os retângulos cinza representam os intervalos de tempo durante os quais uma ação pode ser executada, desde que as restrições de ordem sejam respeitadas. A parte desocupada de um retângulo cinza indica uma folga de tempo. O caminho crítico é aquele cuja duração total é a mais longa; o caminho é “crítico” porque determina a duração de todo o plano — encurtar outros caminhos não encurta o plano como um todo, mas o adiamento do início de qualquer ação no caminho crítico retarda o plano todo. As ações que estão fora do caminho crítico têm uma janela de tempo em que podem ser executadas. A janela é especificada em termos da hora de início mais cedo possível, ES (earliest start), e da mais tardia possível, LS (latest start). O intervalo entre ES e LS é conhecido como folga de uma ação. Podemos visualizar que todo o plano vai levar 85 minutos, que cada ação no processo superior tem 15 minutos de folga e que cada ação no caminho crítico não tem folga (por definição). Juntos, o ES e o LS determinam o tempo para todas as ações, constituindo o escalonamento (plano com indicação de tempos e recursos) para o problema.

Referências:

Livro: Inteligência Artificial

Autor: Peter Norvig