4
Modelos Propostos para Otimização de Planejamentos com Restrições de Precedência

4.1
Representação com Algoritmos Genéticos

Para definir a representação de um modelo para problemas de planejamento com restrições de precedência, partiu-se do princípio de que esta representação deveria ser escolhida de tal modo que soluções ilegais não fossem geradas durante o processo evolucionário. Este princípio é fundamental para o desempenho do sistema uma vez que evita a penalização, descarte ou reparo de cromossomas gerados pelo algoritmo evolucionário.

Neste sentido, o modelo de representação baseado em ordem (em particular a representação por caminho que é a mais utilizada) conforme apresentado anteriormente, não satisfaz esse desejo de forma trivial dado que, a existência das precedências faz com que algumas soluções não sejam válidas. Suponhamos por exemplo um problema com 3 tarefas diferentes: A, B e C e que a tarefa B deva ser executada obrigatoriamente antes da tarefa C. Neste caso, teremos um conjunto de soluções válidas (A,B,C), (B,A,C), (B,C,A) e um conjunto de soluções inválidas (A,C,B), (C,A,B), (C,B,A). Deste modo, ou a função de avaliação descarta estas soluções ou encontra uma maneira de torná-las válidas.

Para contornar este problema, criou-se um modelo baseado em grafos capaz de resolver esta particularidade. Este modelo foi inspirado em um modelo para solução do problema do caixeiro viajante com restrições de precedência (Moon et al, 2002) que é detalhado a seguir.

A abordagem utilizada baseia-se em uma ordenação topológica (*Topological Sort* – TS) que habilita o AG a gerar sempre soluções válidas durante a evolução. A ordenação topológica é uma ordenação de vértices em um grafo dirigido tal que, se existe um caminho entre um vértice v_i e um vértice v_j então v_j aparece depois de v_i na ordenação. Um exemplo de um grafo válido é mostrado na figura a seguir:

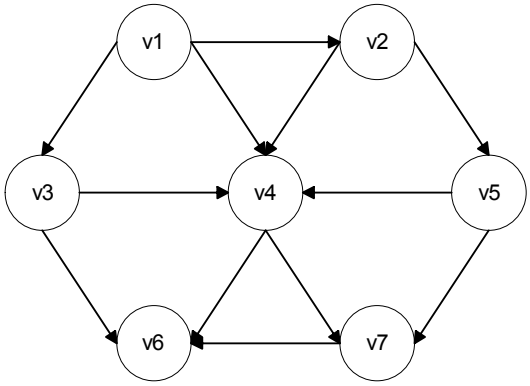

Figura 8 – Exemplo de grafo definindo precedências

O procedimento para se ordenar os vértices (gerando um caminho de vértices ordenados) é trivial e consiste primeiramente em selecionar e armazenar qualquer vértice que não tenha nenhuma seta de entrada (no exemplo da Figura 8 existe apenas o vértice v_1). Feito isto, o vértice é eliminado do grafo bem como as setas que partem dele. Em termos de planejamento, pode-se pensar nos vértices como as tarefas a planejar e nas setas como as relações de precedência entre as mesmas. Desta forma, o caminho $<v_i,v_j>$ de um grafo dirigido indica que a tarefa v_i deve ser executada antes da tarefa v_j. Fica claro também que a ordenação topológica só é viável se não existir um ciclo no grafo.

O outro ponto a se considerar é como selecionar um vértice (tarefa) quando mais de um deles não têm precedentes. É neste ponto que entra a definição da representação do cromossoma. O que se precisa é uma lista que defina uma estrutura, representando as prioridades de execução ou planejamento das tarefas dado que elas não tenham precedentes. Define-se portanto um cromossoma para um grafo de n vértices como um vetor de n posições. Cada uma das posições armazena um valor único entre 1 e n que determina a prioridade de cada uma das tarefas (Moon et al, 2002). Na tabela a seguir mostra-se um exemplo de uma lista para o grafo com 7 tarefas da Figura 8:

Tarefa (Vértice)	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Prioridade	5	1	7	2	4	6	3

Tabela 3 – Lista de prioridades para um grafo com 7 tarefas

A partir daí pode-se usar um algoritmo como o descrito abaixo para se gerar um planejamento válido:

procedure: gerar planejamento válido
    input: grafo dirigido
    while (existir vértice)
        if (todo vértice tem um precedente) **then** grafo inválido: **stop**
        else selecione um vértice *v* com a maior prioridade entre os que não têm precedentes
        planejar *v*
        remover *v* do grafo e todos as ligações que partem dele
    end while
end procedure

Figura 9 – Algoritmo para gerar um planejamento válido a partir de uma lista de prioridades

Para o exemplo de grafo mostrado na figura 8 tem-se então os seguintes passos na geração do planejamento: a primeira tarefa a ser planejada é a tarefa v_1 já que esta tarefa é a única que não possui precedentes. Após planejar a tarefa, remove-se a mesma do grafo bem como as ligações $<v_1,v_2>$, $<v_1,v_3>$,$<v_1,v_4>$. O grafo resultante assume a seguinte forma:

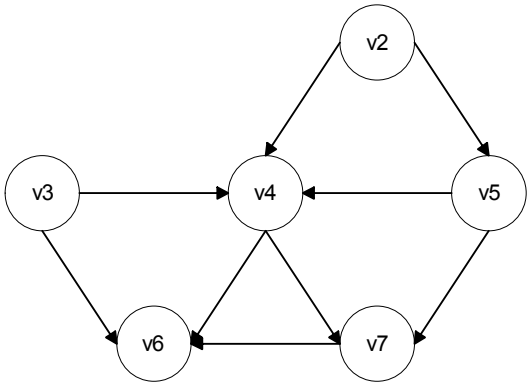

Figura 10 – Grafo após o planejamento de uma tarefa

Pode-se neste momento selecionar o vértice v_2 ou o vértice v_3. No entanto, de acordo com a Tabela 3 o vértice v_3 tem prioridade maior do que o vértice v_2 e

portanto deve ser retirado para planejamento antes. Remove-se também as ligações correspondentes e o grafo agora tem a seguinte forma:

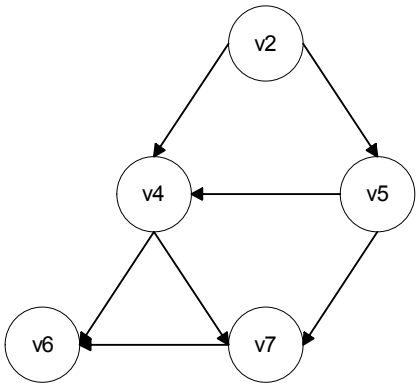

Figura 11 – Grafo após o planejamento de duas tarefas

Seguindo-se este procedimento até não mais existir vértices no grafo, tem-se a seqüência $<v_1, v_3, v_2, v_5, v_4, v_7, v_6>$. Esta seqüência é a única que pode ser gerada a partir deste grafo (dado o vetor de prioridades) e é uma seqüência válida já que respeita todas as precedências.

Para os problemas de planejamento, uma pequena mudança foi introduzida no modelo original, utilizado para a otimização de problemas do caixeiro viajante, com o objetivo de melhorar o desempenho. Da maneira mostrada acima, para se identificar a tarefa de maior prioridade, deve-se percorrer todas as posições do cromossoma verificando-se se existem precedentes para cada tarefa. Para problemas com muitas tarefas, esta busca pode-se tornar computacionalmente custosa. Sendo assim, inverteu-se o significado do cromossoma, que passou a ser um vetor de *n* posições, onde cada posição representa uma prioridade sendo que a primeira posição tem a maior prioridade e a última posição tem a menor prioridade. Cada uma destas posições por sua vez armazena uma das tarefas a ser executada. Sendo assim, percorre-se o cromossoma da esquerda para a direita e ao se encontrar uma tarefa sem precedentes pode-se com segurança saber que esta tarefa é a de maior prioridade. O cromossoma mostrado na tabela 3 assume então a seguinte forma:

Tarefa (Vértice)	v_3	v_6	v_1	v_5	v_7	v_4	v_2
Prioridade	1	2	3	4	5	6	7

Tabela 4 – Cromossoma com 7 tarefas

Deve-se notar que a tarefa que tinha a maior prioridade na Tabela 3, no caso a tarefa v_3 com prioridade 7, também tem a maior prioridade na Tabela 4, mas o valor que indica esta prioridade agora é 1.

A função de avaliação deste modelo de representação por sua vez, contém um decodificador que, a partir do cromossoma e do grafo de precedência é capaz de gerar (construir) um planejamento válido para ser avaliado de acordo com os objetivos que se deseja alcançar. Em geral ao se construir um planejamento, deseja-se não só otimizar o planejamento quanto ao momento em que uma tarefa é realizada, como também o recurso que ela vai utilizar. A seleção de recursos e do instante em que a tarefa deve ser planejada a partir do cromossoma decodificado, são em geral heurísticas e variam de problema para problema. No modelo aqui poderia-se adotar uma heurística simples de seleção de recursos – uma lista fixa e ordenada com os recursos que podem ser utilizados para executar cada tipo de tarefa que é percorrida até se encontrar um recurso que esteja disponível para executar determinada tarefa. Caso não encontre nenhum recurso disponível, o algoritmo tenta alocar o primeiro recurso que estiver disponível para a tarefa (e portanto, adiando o início e o fim da tarefa de modo que esta se adapte ao tempo livre do recurso). Esta heurística é mostrada no algoritmo a seguir:

```
procedure selecionar recurso
    input lista ordenada com os recursos que podem ser usados para a tarefa
    i ← 0
    while ( true )
        repeat
            if (lista [i] está disponível do início ao fim da tarefa)
                usar recurso lista [i]
                terminar
            else
                i ← i + 1
        until (i = tamanho (lista))
        deslocar inícioTarefa para o primeiro momento em que um dos
        recursos da lista estiver livre
        deslocar fimTarefa para a frente, da mesma quantidade que o início
        da tarefa foi deslocado
    end while
end procedure
```

Figura 12 – Algoritmo para selecionar o recurso para execução da tarefa

Finalmente, com relação aos operadores genéticos para este modelo de representação de cromossoma, pode-se usar os operadores genéticos convencionais para problemas baseados em ordem com representação de caminho. Estes operadores (de crossover – PMX, OX, CX – e de mutação – Swap, PI, RLM, RRM) foram descritos detalhadamente no capítulo 2.

4.2 Representação com Algoritmos Co-Evolucionários Cooperativos

Uma das limitações do modelo de representação com algoritmos genéticos convencionais reside no fato, já descrito no capítulo 3, de que os diversos subcomponentes da solução não podem interagir entre si. Isto é notável nos problemas de planejamento que, em geral, trabalham com uma gama de variáveis. Uma dessas limitações por exemplo, consiste no fato de se estar definindo de forma unificada, através da codificação do cromossoma, informações sobre

quando planejar a tarefa e com qual recurso executá-la. Isto cria uma dificuldade extra para o AG já que nem sempre uma boa solução em termos do tempo é bem avaliada, uma vez que, a parte da solução que diz respeito aos recursos utilizados pode ser de má qualidade. Além disso, dependendo das heurísticas utilizadas para se alocar o tempo e os recursos para realizar a tarefa, mesmo que as informações de tempo e recurso codificadas no cromossoma constituam, separadamente, boas soluções, ao avaliá-las de forma unificada pode-se ter uma solução ruim.

Para se contornar este problema de seleção de tempo e recurso, propõe-se um algoritmo genético co-evolucionário cooperativo que seja capaz de solucionar estas situações.

Isto é feito criando-se duas espécies distintas. A primeira com a mesma forma do modelo descrito para o algoritmo genético convencional, que será utilizada para selecionar o momento no qual a tarefa deve ser planejada. A segunda, irá codificar os recursos que devem ser utilizados para se realizar cada uma das tarefas ao invés de se delegar esta tarefa a uma heurística qualquer, porém utilizando o mesmo algoritmo descrito na Figura 12. Em outras palavras, o que se está fazendo neste caso é, ao invés de utilizar uma lista fixa e ordenada de recursos, permite-se que uma outra espécie seja responsável por evoluir uma lista ordenada de recursos que melhor se adapte a solução do problema, cooperando com a espécie responsável por selecionar quando a tarefa será executada. O diagrama a seguir mostra a configuração sugerida:

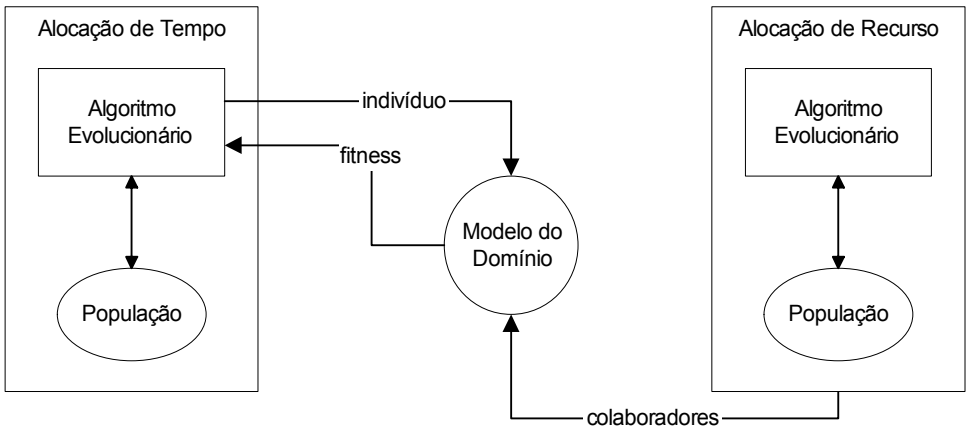

Figura 13 – Modelo co-evolucionário para planejamento

Os indivíduos da população da espécie "Alocação de Tempo" tem a forma já descrita para o algoritmo genético convencional, ou seja, o cromossoma contém

uma lista de prioridades associadas a cada uma das tarefas que deve ser realizada (Tabela 4).

Por sua vez, a população da espécie "Alocação de Recurso" contém um cromossoma com uma complexidade um pouco maior, e pode ser descrito da seguinte forma: cada gene está associado a uma tarefa, da mesma forma que o cromossoma da população da espécie "Alocação de Tempo". No entanto, cada gene contém uma lista dos recursos que podem ser utilizados para realizar aquela tarefa, sendo que a ordem na qual essa lista aparece no gene indica a prioridade na qual se deve tentar utilizar aquele recurso, como se pode ver na figura 14:

Tarefas	A	B	C	D	E	F	G
	1	1	3	4	4	6	6
Recursos	2	2		5	5	7	
		3			6		

Figura 14 – Cromossoma representando os recursos para cada tarefa

Através desta figura então, conclui-se que para a tarefa A, pode-se utilizar os recursos 1 e 2, para a tarefa B os recursos 1, 2 e 3, para a tarefa C o recurso 3 e assim por diante.

O algoritmo para construir a solução é descrito na figura 15. É importante notar que o planejamento da tarefa é feito pelo mesmo algoritmo utilizado no AG com uma única população. A única diferença é que a seleção dos recursos é feita por um procedimento a parte e que já foi descrito na Figura 12 (e neste caso, como já foi dito, ao invés da lista fixa e ordenada, usa-se o cromossoma descrito acima):

procedure decodificador

cromossomaT ← input cromossoma espécie "Alocação Tempo"

cromossomaR ← input cromossoma espécie "Alocação Recurso"

grafo ← input grafo com restrições de precedência

j ← 0

while (*j* < tamanho (*cromossomaT*))

planejar tarefa *j* usando *cromossomaT* e *grafo*

selecionar recurso usando *cromossomaR*

end while

end procedure

Figura 15 – Algoritmo decodificador

É importante notar que a função de avaliação que contém o decodificador de soluções está embutida no modelo do domínio e não nos algoritmos genéticos de cada espécie. Tudo o que a função de avaliação da espécie faz é requisitar ao modelo de domínio que avalie um determinado indivíduo em colaboração com outros e esperar que esta função de avaliação retorne um *fitness* para ela, utilizando um dos métodos descritos no capítulo anterior. Portanto, a heurística é feita sobre a colaboração (ou seja, a solução completa formada por um indivíduo de cada espécie) e não sobre um indivíduo de uma única espécie.

Os operadores genéticos utilizados para a espécie "Alocação de Tempo" são os mesmos utilizados no algoritmo genético convencional. Os operadores utilizados para a espécie "Alocação de Recursos" também são os mesmos, mas eles atuam no conteúdo dos genes e não nos genes propriamente ditos. Ou seja, o crossover é realizado entre cada uma das tarefas, mas em apenas uma de cada vez, de modo que não são trocados recursos entre tarefas diferentes. Em outras palavras, ao fazer o crossover entre dois genitores diferentes, faz-se primeiro o crossover entre a lista de recursos da primeira tarefa do primeiro genitor e a lista de recursos da primeira tarefa do segundo genitor, depois, entre a segunda tarefa do primeiro e do segundo genitor, e assim por diante. Do mesmo modo, a mutação atua em cada uma das listas de recursos de cada uma das tarefas, nunca trocando material genético de uma tarefa com a outra. Estas duas operações podem ser vistas na figura 16:

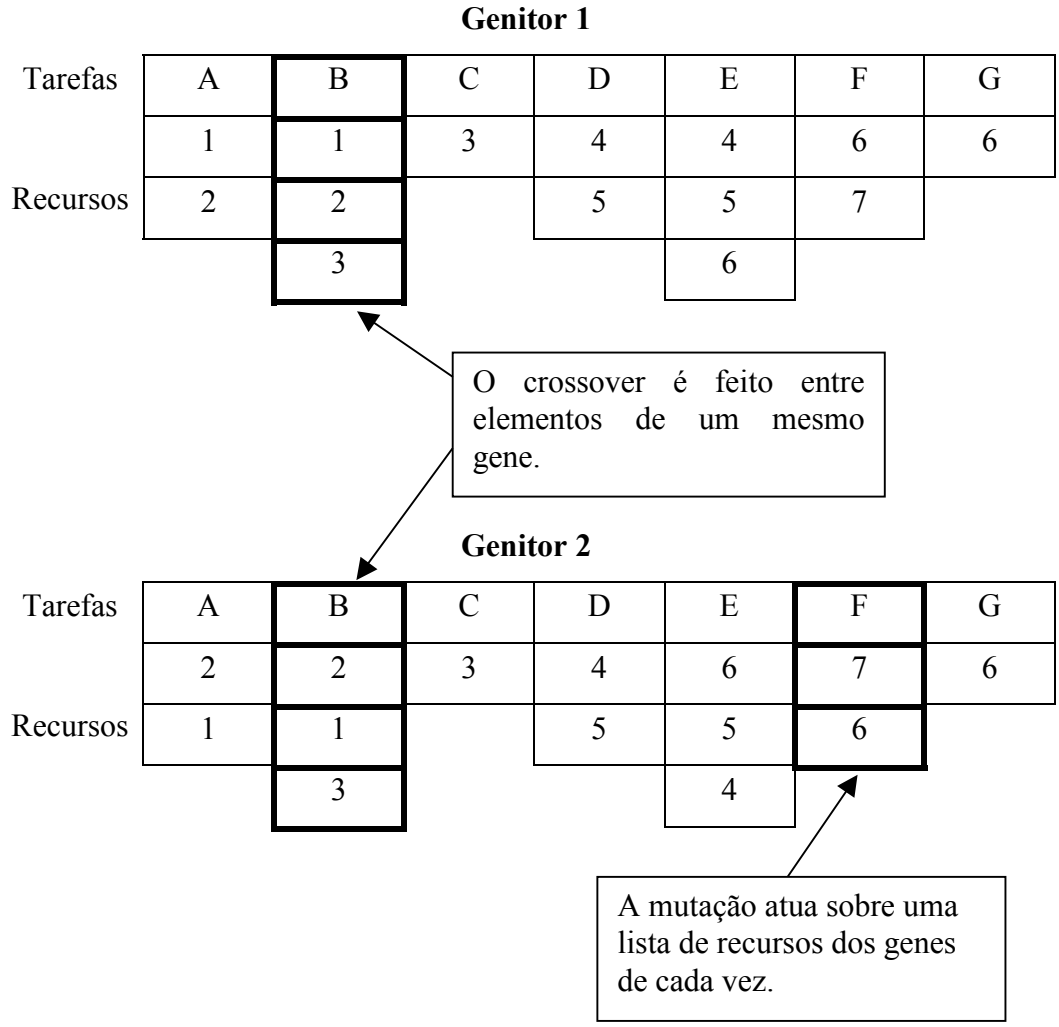

Figura 16 – Atuação do crossover e da mutação sobre o cromossoma

No próximo capítulo, será mostrado o uso deste modelo de representação em um estudo de caso em que se deseja otimizar o planejamento de descarga e embarque de minério em um porto fictício.