Inteligência Computacional para Otimização

Marcone Jamilson Freitas Souza, Departamento de Computação, Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto, 35400-000 Ouro Preto, MG. Homepage: http://www.decom.ufop.br/prof/marcone, E-mail: marcone@iceb.ufop.br

1. Heurísticas Construtivas

Uma heurística construtiva tem por objetivo construir uma solução, elemento por elemento. A forma de escolha de cada elemento a ser inserido a cada passo, varia de acordo com a heurística, a qual, por sua vez, depende do problema abordado.

Mostraremos uma heurística construtiva utilizando o Problema da Mochila como exemplo.

Seja, então, uma mochila de capacidade $b = 23$ e os 5 objetos da tabela abaixo, com os respectivos pesos e benefícios.

Objeto (j)	1	2	3	4	5
Peso (w_j)	4	5	7	9	6
Benefício (p_j)	2	2	3	4	4

Construamos uma solução para esse problema usando a seguinte idéia: adicionemos à mochila a cada passo, o objeto mais valioso por unidade de peso e que não ultrapasse a capacidade da mochila. Reordenando os objetos de acordo com a relação p_j/w_j, obtemos:

Objeto (j)	5	1	4	3	2
Peso (w_j)	6	4	9	7	5
Benefício (p_j)	4	2	4	3	2
(p_j/w_j)	0.67	0.50	0.44	0.43	0.40

Representemos uma solução s por um vetor binário de n posições.

Passo 1 : Adicionemos, primeiramente, o objeto 5, que tem a maior relação p_j/w_j
$s = (00001)^t$
$f(s) = 4$
Peso corrente da mochila = 6 < b = 23

Passo 2 : Adicionemos, agora, o objeto 1, que tem a segunda maior relação p_j/w_j
$s = (10001)^t$
$f(s) = 6$
Peso corrente da mochila = 10 < b = 23

Passo 3 : Adicionemos, agora, o objeto 4, que tem a terceira maior relação p_j/w_j
$s = (10011)^t$
$f(s) = 10$
Peso corrente da mochila = 19 < b = 23

Passo 4 : O objeto a ser alocado agora seria o terceiro. No entanto, esta alocação faria superar a capacidade da mochila. Neste caso, devemos tentar alocar o próximo objeto com a maior relação p_j/w_j, que é o objeto 1. Como também a alocação desse objeto faria superar a capacidade da mochila e não há mais objetos, concluímos que a solução anterior é a solução final, isto é: $s^* = (10011)^t$ com $f(s^*) = 10$.

Deve ser observado que poderíamos utilizar uma outra idéia de construção para resolver esse problema. Por exemplo, poderíamos utilizar a seguinte idéia: Adicionar a cada passo o objeto que proporcionar o maior benefício e que não ultrapassar a capacidade da mochila. Resolva o problema dado por esta heurística e verifique que a solução final obtida foi diferente! Esses dois exemplos mostram que diferentes heurísticas conduzem a diferentes soluções finais.

Uma outra forma muito comum de se gerar uma solução inicial é construí-la de maneira aleatória. Isto é, a cada passo um elemento a ser inserido na solução é aleatoriamente selecionado do conjunto de elementos ainda não selecionados. A grande vantagem dessa metodologia reside na simplicidade de implementação. A desvantagem é a qualidade baixa da solução final produzida, o que requerirá um esforço maior na fase de refinamento.

2. Heurísticas de Refinamento

As heurísticas de refinamento em problemas de otimização, também chamadas de técnicas de busca local, constituem uma família de técnicas baseadas na noção de vizinhança. Mais especificamente, seja S o espaço de pesquisa de um problema de otimização e f a função objetivo a minimizar. A função N, a qual depende da estrutura do problema tratado, associa a cada solução $s \in S$, sua vizinhança $N(S) \subseteq S$. Cada solução $s' \in N(s)$ é chamada de vizinho de s. Denomina-se *movimento* a modificação m que transforma uma solução s em outra, s', que esteja em sua vizinhança. Representa-se essa operação por $s' \leftarrow s \oplus m$.

Em linhas gerais, essa classe de heurística parte de uma solução inicial qualquer (a qual pode ser obtida por uma heurística construtiva ou então gerada aleatoriamente) e caminha, a cada iteração, de vizinho para vizinho de acordo com a definição de vizinhança adotada. As heurísticas clássicas de refinamento são as seguintes:

2..1 Método da Descida/Subida (Descent/Uphill Method)

A idéia dessa técnica é partir de uma solução inicial qualquer e a cada passo analisar todos os seus possíveis vizinhos, movendo somente para aquele que representar uma melhora na função de avaliação atual.

Para o problema da mochila, consideremos uma solução s sendo representada por um vetor binário de n posições e como movimento m a troca do valor de um bit. Assim, a vizinhança de uma solução s, e se escreve $N(s)$ é o conjunto de todos os vizinhos s' que diferem de s pelo valor de um bit. Formalmente, representamos $N(s) = \{s' : s' \leftarrow s \oplus m\}$, onde m significa a troca de um bit. É necessário, agora, definir uma função de avaliação. Uma função de avaliação apropriada seria:

$$f(s) = \sum_{j=1}^{n} p_j s_j - \alpha \times \max\{0, \sum_{j=1}^{n} w_j - b\} \tag{2.1}$$

sendo α uma penalidade, por exemplo, $\alpha = \sum_{j=1}^{n} w_j = 31$.

Observe que o objetivo da segunda parcela desta função de avaliação é penalizar a colocação na mochila de objetos que ultrapassam a capacidade da mochila.

Apliquemos essa heurística ao problema dado:

Passo 0 : Seja uma solução inicial qualquer, por exemplo:
$s = (01010)^t$
$f(s) = 6$
Peso corrente da mochila = 14

Passo 1 : Devemos, agora, analisar todos os vizinhos de s e calcular a função de avaliação deles usando a expressão 2.1.

Vizinhos de s	Peso dos vizinhos de s	Benefício dos vizinhos de s	$f(s)$
$(\mathbf{1}1010)^t$	18	8	8
$(0\mathbf{0}010)^t$	9	4	4
$(01\mathbf{1}10)^t$	21	9	9
$(010\mathbf{0}0)^t$	5	2	2
$(0101\mathbf{1})^t$	20	10	10

Melhor vizinho: $s' = (01011)^t$
$f(s') = 10$
Como $f(s') > f(s)$ então $s \leftarrow s'$, isto é:
$s = (01011)^t$

Passo 2 : Determinemos, agora, o melhor vizinho de $s = (01011)^t$:

Vizinhos de s	Peso dos vizinhos de s	Benefício dos vizinhos de s	$f(s)$
$(\mathbf{1}1011)^t$	24	12	-19
$(0\mathbf{0}011)^t$	15	8	8
$(01\mathbf{1}11)^t$	27	13	-111
$(010\mathbf{0}1)^t$	11	6	6
$(0101\mathbf{0})^t$	14	6	6

Melhor vizinho: $s' = (00011)^t$
$f(s') = 8$
Como $f(s')$ é pior que $f(s)$ então PARE. A solução anterior é um ótimo local, isto é, o método da descida retorna $s^* = (01011)^t$, com $f(s^*) = 10$ como solução final.

É importante observar que diferentes soluções iniciais conduzem, na maioria das vezes, a diferentes soluções finais. Faça essa verificação!