

MO824 - Atividade Busca Tabu

Rebecca Messias (123456)
Ieremies Romero (217938)
Renan Franco (123456)

Descrição do problema

A partir do problema da *Maximum Quadratic Binary Function* (MAX-QBF), proposto por [qbf], o problema de maximização da função $f : \mathbb{B}^{|x|} \rightarrow \mathbb{R}$, onde $|x|$ é a dimensão do problema, descrita como

$$f(x_1, \dots, x_n) = \sum_{i=1}^n \sum_{j=1}^n a_{i,j} x_i x_j,$$

em que $a_{i,j} \in \mathbb{R}$, $j = 1, \dots, n$ são os coeficientes da função.

A este problema adicionamos a restrição

$$\sum_{i=1}^n w_i x_i \leq W,$$

onde w_i é dito o peso da variável x_i . Nosso objetivo nessa nova versão chamada *Maximum Knapsack Quadratic Binary Function* (MAX-KQBF) é maximizar a função, tal que a soma dos pesos das variáveis na solução não exceda W .

Nesse problema, nossas variáveis de decisão consistem em tornar ou não um certo x_i para valor 1 ou não. Isso será representado no código como inserir ou não na solução.

Durante esse relatório, nos referiremos a $c(x_i)$ como o custo que o elemento x_i incumbe a atual solução. Caso x_i já esteja na solução, ele é a diferença no valor da função objetivo da solução atual e sem ele. Caso x_i não pertença à solução, $c(x_i)$ é a diferença no valor da função objetivo da solução atual e com ele.

Metodologia

Heurística construtiva

Neste projeto, utilizamos como heurística padrão, um algoritmo muito similar ao proposto por [rezende19_grasp].

Iniciamos com uma lista de candidatos CL composta por todos os elementos podem ser adicionados a solução e que, se o feito, infringirão na restrição de peso. A cada interação, atualizamos CL com o mesmo critério, removendo aqueles que não podem mais serem adicionados.

Baseado no custo adicional que cada candidata incumbirá à solução quando inserida, determinamos quais os maiores (*maxCost*) e menores (*minCost*) custos e, baseado no parâmetro α , determinamos a lista de candidatos restritos de forma gulosa por

$$RCL = \{cand : c(cand) \leq minCost + \alpha (maxCost - minCost)\}.$$

Por fim, concluímos a interação da heurística realizando o passo aleatório: escolhemos um elemento de RCL aleatoriamente a ser inserido na nossa solução. Observe que o parâmetro α determina o espaço amostral que teremos para retirar nosso elemento aleatório. Caso este seja pequeno de mais, realizamos apenas uma heurística gulosa, caso seja grande de mais, temos um algoritmo extremamente aleatório. Nesse experimento trabalhamos com dois valores de α : $\alpha_1 = 0.05$ e $\alpha_2 = 0.17$.

Para a heurística construtiva, consideramos como critério de parada quando a nossa lista de candidatas CL se tornar vazia ou nenhuma inserção que possamos fazer irá melhorar nossa função objetivo.

Lista tabu

Estrutura e atualização.

Critério de aspiração

Operadores de busca local

Métodos de busca

Critérios de parada

Estratégias alternativas

Resultados