Programação Inteira
Geração de Colunas em PI

Haroldo Gambini Santos

Universidade Federal de Ouro Preto

30 de agosto de 2011

Notas

Motivação

- frequentemente, a modelagem precisa dos problemas de programação linear exige um conjunto muito grande de variáveis
- no caso de PI, muitas as vezes as melhores formulações (fortes) são aquelas que incluem um número exponencial de variáveis
- para a resolução desses problemas é necessário **somente**:
  - restrições apertadas
    - restrições com valores duais positivos
  - variáveis básicas na solução ótima
    - variáveis com valores positivos na otimalidade
- como descobrir essas variáveis e restrições ?

Notas

Geração Implícita de Colunas

Delayed Column Generation

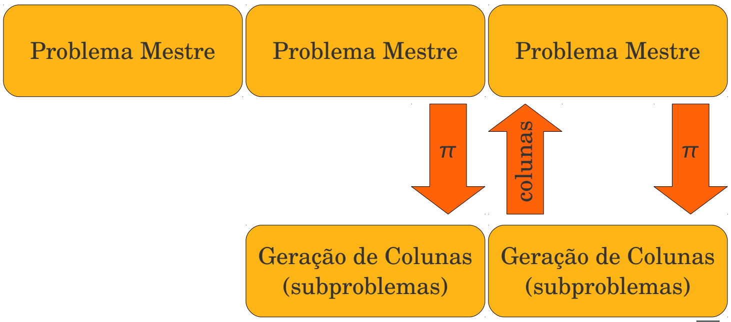
1. Resolva o programa linear $LP(J)$:

$$\begin{aligned} min : \sum_{j \in J} c_j x_j \\ Ax \geq b \\ x \geq 0 \end{aligned}$$

2. ***Pricing***: considerando o valor das variáveis duais π
   descubra se existe algum $j \notin J$ tal que
   $c_j - \pi A_j \leq 0$
   (*custo reduzido*)
   se existir volte para 1
   caso contrário a solução corrente é ótima

Notas

Decomposição



Notas

Decomposição de *Dantzig-Wolfe*

- uso comum de geração de colunas: separar o conjunto de restrições “fáceis” do resto das restrições do problema (as restrições “complicadoras”)
  - as restrições fáceis podem envolver uma pequena porção das variáveis do problema
  - essas restrições podem apresentar uma estrutura especial, como a de algum problema em grafos para o qual existam algoritmos rápidos de resolução
- desse modo, pode-se resolver separadamente as restrições fáceis

Notas

Exemplo: O Problema da Alocação Generalizada

Entrada

- n tarefas
- m máquinas
- $l_{i,j}$ lucro associado a realização da tarefa i na máquina j
- r_{ij} quantidade de recursos que a tarefa i consome na máquina j
- c_j capacidade da máquina j

Entrada

- n tarefas
- m máquinas
- l_{ij} lucro tarefa i na máquina j
- r_{ij} recursos tar. i consome máq. j
- c_j capacidade da máquina j

Formulação padrão

$$Max: \sum_{i=1,\ldots,n} \sum_{j=1,\ldots,m} l_{ij}x_{ij}$$

Notas

Ex.: Decomposição DW para o PAG

$$Max: \sum_{i=1,\dots,n} \sum_{j=1,\dots,m} l_{ij} x_{ij}$$

$$S.a.: \sum_{j=1,\dots,m} x_{ij} = 1 \; \forall i \in \{1, \dots, n\}$$

$$\sum_{i=1,\dots,n} r_{ij} x_{ij} \leq c_j \; \forall j \in \{1, \dots, m\}$$

Problema Mestre restrição de associação

Subproblemas restrição de capacidade

Notas

Coluna = Padrão

Considere, que para cada máquina são gerados todos os padrões de alocação de tarefas que respeitam a capacidade da mesma. Cada padrão irá corresponder a uma variável (coluna).

Colunas / Tarefas	1	2	...		...	
1	1	0	...	1	...	1
2	0	1	...	1	...	1
3	0	0	...	0	...	1
4	0	0	...	0	...	0
5	0	0	...	1	...	0

(exemplo com 5 tarefas)

Notas

DW para o PAG

Colunas / Tarefas	1	2	...		...	
1	1	0	...	1	...	1
2	0	1	...	1	...	1
3	0	0	...	0	...	1
4	0	0	...	0	...	0
5	0	0	...	1	...	0

Para uma máquina j onde existem K_j padrões factíveis de alocação de tarefas, criamos a variável binária

$$\lambda_k^j = \begin{cases} 1 & \text{se a máquina } j \text{ irá processar as tarefas do padrão } k \\ 0 & \text{c.c.} \end{cases}$$

Conteúdo da coluna k especificado por $y_i^j = (y_1^j, \dots, y_n^j)$, $y_i^j \in \{0, 1\}$. Cada coluna deve satisfazer:

$$\sum_{i=1}^{n} r_{ij} y_i^j \leq c_j$$

Notas

O Problema de *Pricing*

Considerando cada máquina, devem-se descobrir quais são os padrões:

$$(y_1^j, \ldots, y_n^j)$$

promissores.

Desse modo, considerando a variável dual π_j referente a máquina j, temos o seguinte problema de *Pricing*:

$$Max.: \sum_{i=1}^{n} (l_{ij} - \pi_j r_{ij}) y_i^j$$

$$s.a.: \sum_{i=1}^{n} r_{ij} y_{ij} \le c_j$$

Notas

Decomposição DW para o PAG

DW Mestre PAG

$$Max.: \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{K_j} l_{ij} y_{ik}^j \lambda_k^j$$

$$\sum_{j=1}^{m} \sum_{k=1}^{K_j} y_{ik}^j \lambda_k^j = 1 \;\; \forall i \in \{1, \ldots, n\}$$

$$\sum_{k=1}^{K_j} \lambda_k^j = 1 \;\; \forall j \in \{1, \ldots, m\}$$

Notas

Decomposição DW para o PAG

- Calcular a relaxação linear para o DW Mestre PAG será provavelmente muito mais demorado do que computar a relaxação da formulação padrão
- A vantagem de utilizar DW Mestre PAG vem do fato do mesmo prover uma formulação mais ***forte*** do que a formulação original
  - soluções fracionárias que não são combinações convexas das soluções 0-1 para as restrições da mochila não são factíveis para DW Mestre PAG
- Desse modo, considerando geração de colunas em Programação Inteira, não se buscam programas lineares mais rápidos de resolver, mas sim programas lineares que ofereçam **limites melhores**
- A relaxação dessas formulações melhores, além de produzir limites mais justos, muitas vezes é mais "Integer Friendly"

Notas