



Branch & Bound – Ramificar e Podar

- Roda sobre a estrutura do *Backtracking*
 - Geralmente ineficiente no pior caso
- Adiciona a *poda* de soluções parciais não promissoras

2

Ramificação

- O *Branching* (ramificação) consiste em dividir um problema em subproblemas menores
- Seja um problema P , dividimos em m subproblemas:
- Tal que: P_1, P_2, \dots, P_m

$$P_1 \cup P_2, \dots, \cup P_m = P$$

3

Branching

- Uma maneira de se criar subproblemas é através da fixação de variáveis discretas.
- Considere uma variável v , de um problema P , cujos valores possíveis sejam $1, \dots, n$.
- Pode-se dividir P em P_1, \dots, P_n , onde em P_i temos a variável v fixada para o seu i -ésimo valor possível.

4

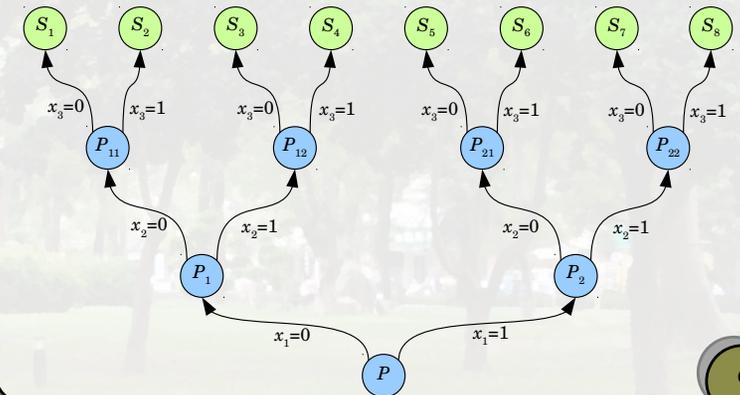
Árvore de Enumeração

- A fixação recursiva de diferentes variáveis cria uma árvore, onde temos:
 - **nós internos**: esses nós representam todas as soluções que podem ser obtidas respeitando as fixações já feitas – são soluções parciais;
 - **folhas**: representam soluções completas.

5

Árvore de Enumeração

- Ex.: Problema com 3 variáveis binárias: x_1, x_2, x_3 .



6

Bounds (Limites)

- usando somente o *branch* temos um *algoritmo exato* que em um número **finito** de passos fornece a solução ótima;
 - extremamente **ineficiente** !
 - 2^n para enumeração de subconjuntos
 - $n!$ para enumeração de permutações
- a chave para melhorar a eficiência do B & B é a poda de algumas sub-árvores através do uso de **limites**.

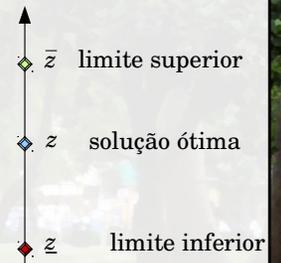
7

Bounds (Limites)

- Considere um problema onde queremos achar o lucro máximo da solução ótima:

$$z = \max(f(x))$$

- Se calcular uma estimativa superior \bar{z} e uma estimativa inferior \underline{z} (ambas válidas), for barato, podemos usar esses limites para podar a árvore de busca por z



8

Exemplo, Problema da Mochila 0/1 - PM01

- Entrada:

n : itens l_i : lucro do item i
 C : capacidade da mochila p_i : peso do item i

- Decisão:

- Objetivo $\begin{cases} 1 & \text{item incluso na solução} \\ 0 & \text{caso contrário} \end{cases}$

- Restrição: $\max \sum_{i=1, \dots, n} l_i x_i$

$$\sum_{i=1, \dots, n} p_i x_i \leq C$$

9

Heurística Rápida

- Ordenar os itens por *densidade de lucro*:

$$d_i = \frac{l_i}{p_i}$$

- Heurística

Enquanto houver algum item que caiba na mochila:
 dentre os itens não incluídos e que cabem na
 mochila inclua o de maior densidade

10

Heurística Rápida - Limite Inferior

- Exemplo: $n=4$ e $C=6$

item	lucro	peso	$\frac{l_i}{p_i}$
1	7	4	1,75
2	4	3	1,33
3	9	5	1,80
4	3	2	1,50

- Solução da heurística:

- $S = (0, 0, 1, 0)$, lucro=9
- Limite inferior válido:** o ótimo é maior ou igual a 9

11

Limite Superior

- O Problema da Mochila fracionário
- Relaxação:**

$$x_i \in \{0, 1\} \quad \longrightarrow \quad x_i \in [0, 1]$$

- Relaxação: remoção de uma ou mais restrições do problema original com o objetivo de deixar o problema mais fácil
- No caso, relaxamos a exigência de valor inteiro para as variável x_i . O problema resultante é o **Problema Fracionário da Mochila - PFM01**

12

Problema Fracionário da Mochila - PFM01

- A solução **ótima** do PFM01 pode ser calculada rapidamente:
 - Ordene itens por densidade
 - Adicione, em ordem, iniciando pelo item de maior densidade cada item, enquanto a capacidade da mochila for suficiente
 - No primeiro item que ficará de fora da solução, caso exista uma capacidade restante, inclua a **maior fração** possível desse item na solução

13

Problema Fracionário da Mochila - PFM01

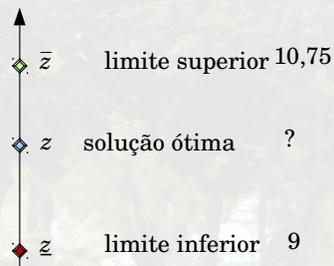
- Exemplo: $n=4$ e $C=6$

item	lucro	peso	$\frac{L_i}{p_i}$
1	7	4	1,75
2	4	3	1,33
3	9	5	1,80
4	3	2	1,50

- Solução:
 - $S = (1/4, 0, 1, 0)$, lucro=10,75
 - **Limite superior válido:** o ótimo é menor ou igual a 10,75

14

Limites Atuais



15

Resolvendo o PM01 – Branch & Bound

- Para encontrar o ótimo de PM01 ainda temos que explorar a árvore de busca
 - Poda:
 - dada uma **solução parcial** e uma **capacidade restante**, o preenchimento dessa capacidade restante através da resolução do PFM01 fornece um **limite superior \bar{z} para o lucro obtido considerando essa solução parcial**, soluções parciais onde \bar{z} é menor do que o lucro de alguma solução válida já conhecida não precisam ser estendidas (poda por limite)
 - soluções parciais com peso maior ou igual a capacidade também não precisam ser estendidas (poda por infactibilidade)

16

Solução Incumbente

- **Melhor solução válida** encontrada até o momento na busca
- No caso do problema da mochila 0/1, qualquer solução parcial que respeita a capacidade da mochila é uma solução válida

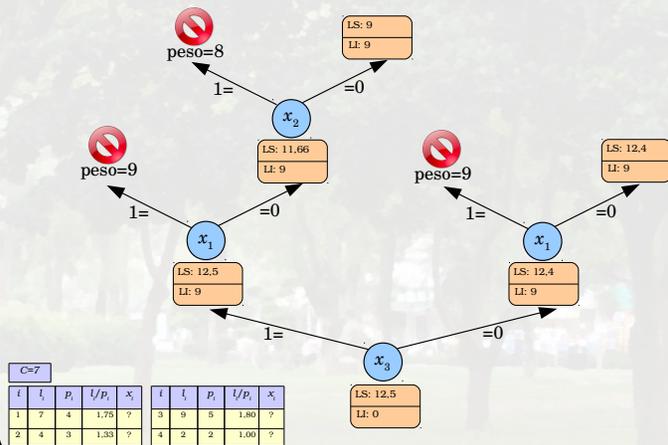
17

Branch & Bound - Desempenho

- Diminuição da árvore:
 - Melhores limites (relaxação boa) e encontrar uma solução incumbente de boa qualidade
- Execução em Tempo Restrito
 - Método pode ser executado até que se encontre uma solução de qualidade desejada
 - Decisões importantes :
 - Ordem das variáveis para o *Branching*
 - Qual fixação de valor explorar primeiro

18

Exemplo



19