Branch-and-Bound para problemas de Otimização Combinatória

Rafael Antônio Marques Gomes

Orientador: Haroldo Gambini Santos

Departamento de Computação
UFOP

26 de julho de 2011

Universidade Federal de Ouro Preto

Universidade Federal de Ouro Preto

O problema geral

Escalonamento

- Distribuir o mais uniformemente possível a carga horária das escalas;
- Atender todas as exigências obrigatórias;
- Atender o máximo de exigências não obrigatórias, como preferências pessoais.

O problema geral

Escalonamento

- Distribuir o mais uniformemente possível a carga horária das escalas;
- Atender todas as exigências obrigatórias;
- Atender o máximo de exigências não obrigatórias, como preferências pessoais.

Motivação

Importância Teórica

- Classe NP-Difícil

Importância Prática

- Satisfação pessoal;
- Melhor qualidade nos serviços prestados;
- Redução de custos.

Universidade Federal de Ouro Preto

Objetivos

Geral

Conseguir o mais rapidamente possível soluções próximas do ótimo para o problema de escalonamento de enfermeiras

Específicos

- Fazer um estudo e apresentar a importância do método *Branch-and-Bound* para o desenvolvimento de uma formulação de Programação Inteira;
- Desenvolver e avaliar técnicas e formulações de programação inteira;
- Testar a formulação proposta com um resolvedor e avaliar os resultados obtidos.

Trabalhos Relacionados

1ª Competição de Escalonamento de Enfermeiras

- No ano de 2010, como um evento paralelo ao PATAT 2010, foi promovida a primeira edição desta competição;
- É um incentivo a pesquisa no campo de escalonamento de enfermeiras e busca estabelecer um problema *benchmark*;
- Busca diminuir a lacuna que existe atualmente entre pesquisa e prática;

Trabalhos Relacionados

1ª Competição de Escalonamento de Enfermeiras

- No ano de 2010, como um evento paralelo ao PATAT 2010, foi promovida a primeira edição desta competição;
- É um incentivo a pesquisa no campo de escalonamento de enfermeiras e busca estabelecer um problema *benchmark*;
- Busca diminuir a lacuna que existe atualmente entre pesquisa e prática;

Trabalhos Relacionados

Vencedores da 1ª Competição

- Ejeção de cadeias e *Branch-and-Price* (Burke e Curtois, 2010);
- Busca-local com técnica de multi-início (Lü e Hao, 2010);
- Metaheurística para o problema de otimização por restrições (Nonobe, 2010);

Fora da competição

- Busca tabu e um algoritmo genético (Dias et al., 2003);

Trabalhos Relacionados

Vencedores da 1ª Competição

- Ejeção de cadeias e *Branch-and-Price* (Burke e Curtois, 2010);
- Busca-local com técnica de multi-início (Lü e Hao, 2010);
- Metaheurística para o problema de otimização por restrições (Nonobe, 2010);

Fora da competição

- Busca tabu e um algoritmo genético (Dias et al., 2003);

Universidade Federal de Ouro Preto

Trabalhos Relacionados

Vencedores da 1ª Competição

- Ejeção de cadeias e *Branch-and-Price* (Burke e Curtois, 2010);
- Busca-local com técnica de multi-início (Lü e Hao, 2010);
- Metaheurística para o problema de otimização por restrições (Nonobe, 2010);

Fora da competição

- Busca tabu e um algoritmo genético (Dias et al., 2003);

Programação Inteira

Programação Linear:

$$max\ 21x_1 + 11x_2$$
$$s.a.\ 7x_1 + 4x_2 <= 13$$
$$x_1, x_2 >= 0$$

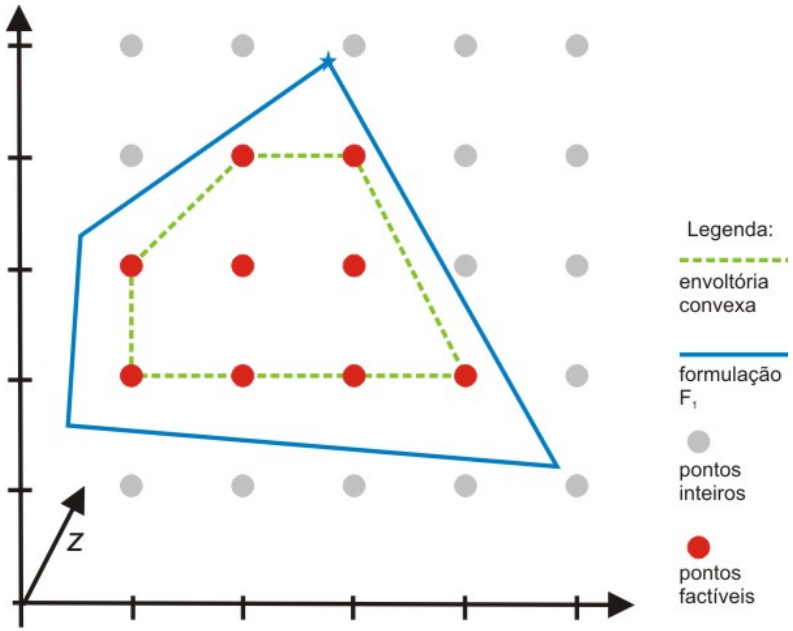
Variáveis Binárias:

$$x_j = \begin{cases} 1, \textit{ se a decisão for sim;} \\ 0, \textit{ caso contrário;} \end{cases}$$

Universidade Federal de Ouro Preto

Envoltória convexa

Restrições de uma formulação devem definir com exatidão a envoltória convexa



- Requer um número muito grande de restrições
- Busca-se formulações que sejam boas aproximações da envoltória convexa

Branch-and-Bound

Idéia Básica

- O algoritmo roda sob a árvore de enumeração das soluções possíveis.
- No pior caso, todas as soluções serão exploradas.
- Na prática, frequentemente vários ramos são **podados** com o uso de limites.

Branch-and-Bound

Idéia Básica

- O algoritmo roda sob a árvore de enumeração das soluções possíveis.
- No pior caso, todas as soluções serão exploradas.
- Na prática, frequentemente vários ramos são **podados** com o uso de limites.

Branch-and-Bound

Branch

- *Branch (Ramificar)* consiste em dividir um problema em problemas menores;
- Uma maneira de se criar subproblemas é através da fixação de variáveis discretas;
- 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;
  - **Folhas:** representam soluções completas.

Universidade Federal de Ouro Preto

Branch-and-Bound

Branch

- *Branch (Ramificar)* consiste em dividir um problema em problemas menores;
- Uma maneira de se criar subproblemas é através da fixação de variáveis discretas;
- 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;
  - **Folhas:** representam soluções completas.

Universidade Federal de Ouro Preto

Bound - Limites

Soluções Parciais

- Métodos como heurísticas ou relaxação podem ser executados em nós internos oferecendo limites **Superior** e **Inferior**
  - **Solução Incumbente:** É a melhor solução encontrada até o momento durante a busca. No caso de Maximização, temos um Limite Inferior.
  - **Relaxação:** Oferece uma solução cujo lucro é melhor ou igual ao da solução ótima do problema original. Em Maximização, temos um Limite Superior.

Bound - Limites

Soluções Parciais

- Métodos como heurísticas ou relaxação podem ser executados em nós internos oferecendo limites **Superior** e **Inferior**
  - **Solução Incumbente:** É a melhor solução encontrada até o momento durante a busca. No caso de Maximização, temos um Limite Inferior.
  - **Relaxação:** Oferece uma solução cujo lucro é melhor ou igual ao da solução ótima do problema original. Em Maximização, temos um Limite Superior.

Poda

Razões para podar um nó

Limite a relaxação (Limite Superior) indica que não há possibilidade de se melhorar a solução incumbente;

Infactibilidade as fixações já feitas induzem a alguma infactibilidade.

Em ambos os casos o nó e todos os seus filhos são podados.

Poda

Razões para podar um nó

Limite a relaxação (Limite Superior) indica que não há possibilidade de se melhorar a solução incumbente;

Infactibilidade as fixações já feitas induzem a alguma infactibilidade.

Em ambos os casos o nó e todos os seus filhos são podados.

Branch and Bound - Exemplo: Problema da Mochila

Incumbente: 11

LS: 12,5
LI: 11

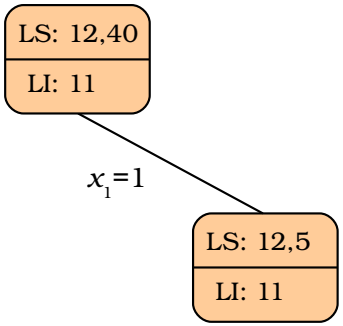
C=7

i	l_i	p_i	l_i/p_i	x_i
1	7	4	1,75	?
2	4	3	1,33	?

i	l_i	p_i	l_i/p_i	x_i
3	9	5	1,80	?
4	2	2	1,00	?

Universidade Federal de Ouro Preto

Branch and Bound - Exemplo: Problema da Mochila



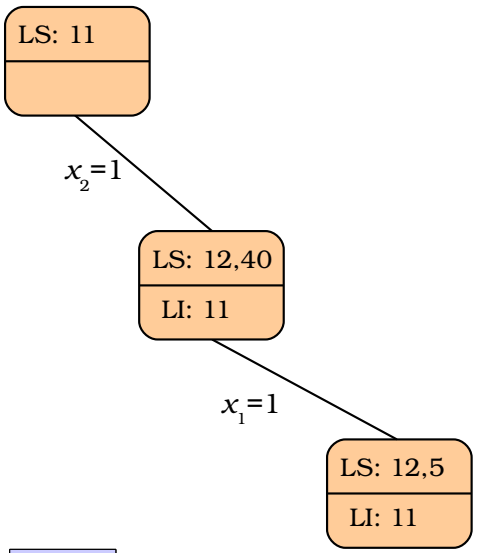
Incumbente: 11

C=7

i	l_i	p_i	l_i/p_i	x_i
1	7	4	1,75	?
2	4	3	1,33	?

i	l_i	p_i	l_i/p_i	x_i
3	9	5	1,80	?
4	2	2	1,00	?

Branch and Bound - Exemplo: Problema da Mochila



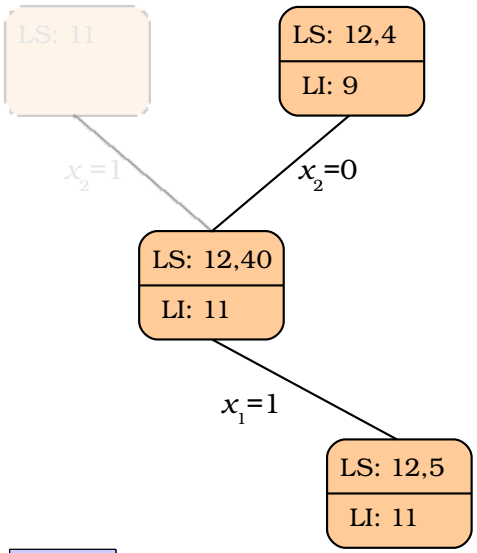
C=7

i	l_i	p_i	l_i/p_i	x_i
1	7	4	1,75	?
2	4	3	1,33	?

i	l_i	p_i	l_i/p_i	x_i
3	9	5	1,80	?
4	2	2	1,00	?

Incumbente: 11
Podado: continuar nesse nó não oferecerá nenhuma solução com custo melhor do que 11.

Branch and Bound - Exemplo: Problema da Mochila



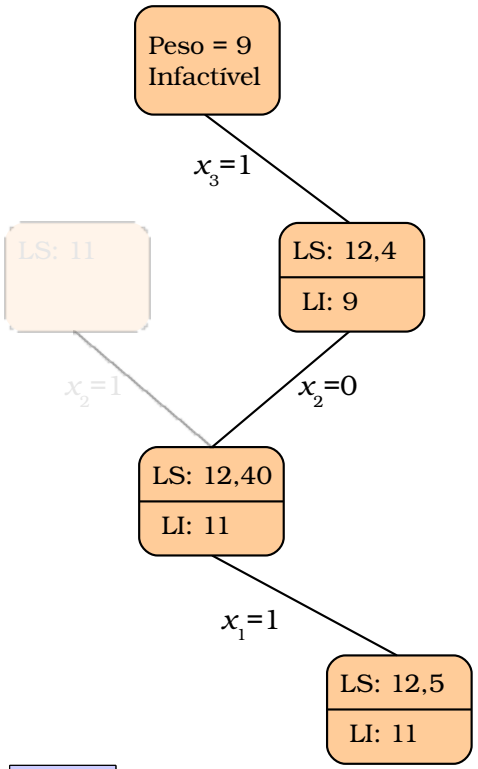
Incumbente: 11

C=7

i	l_i	p_i	l_i/p_i	x_i
1	7	4	1,75	?
2	4	3	1,33	?

i	l_i	p_i	l_i/p_i	x_i
3	9	5	1,80	?
4	2	2	1,00	?

Branch and Bound - Exemplo: Problema da Mochila



C=7

i	l_i	p_i	l_i/p_i	x_i
1	7	4	1,75	?
2	4	3	1,33	?

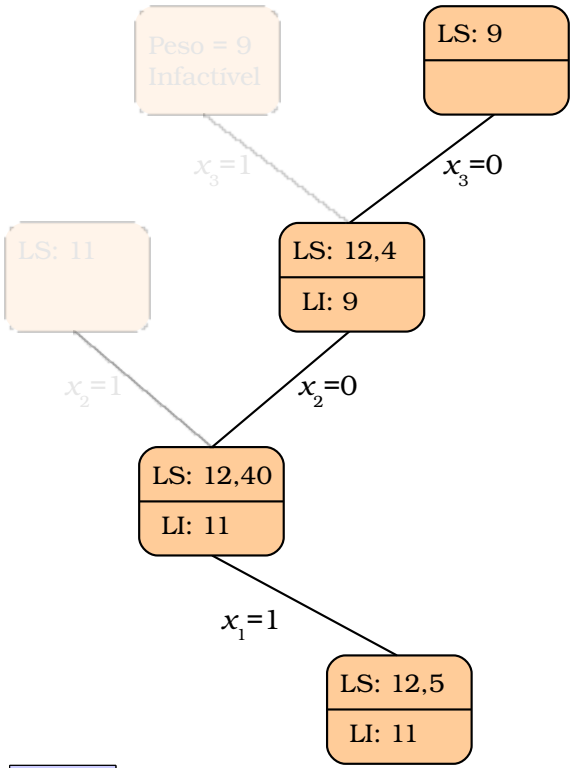
i	l_i	p_i	l_i/p_i	x_i
3	9	5	1,80	?
4	2	2	1,00	?

Incumbente: 11

Podado:

Infactibilidade.

Branch and Bound - Exemplo: Problema da Mochila



C=7

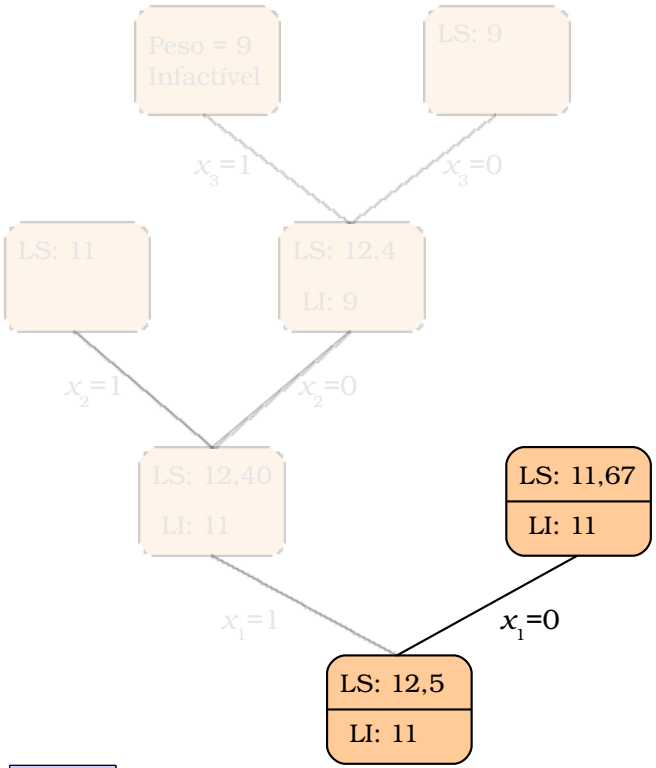
i	l_i	p_i	l_i/p_i	x_i
1	7	4	1,75	?
2	4	3	1,33	?

i	l_i	p_i	l_i/p_i	x_i
3	9	5	1,80	?
4	2	2	1,00	?

Incumbente: 11

Podado por Limite.

Branch and Bound - Exemplo: Problema da Mochila



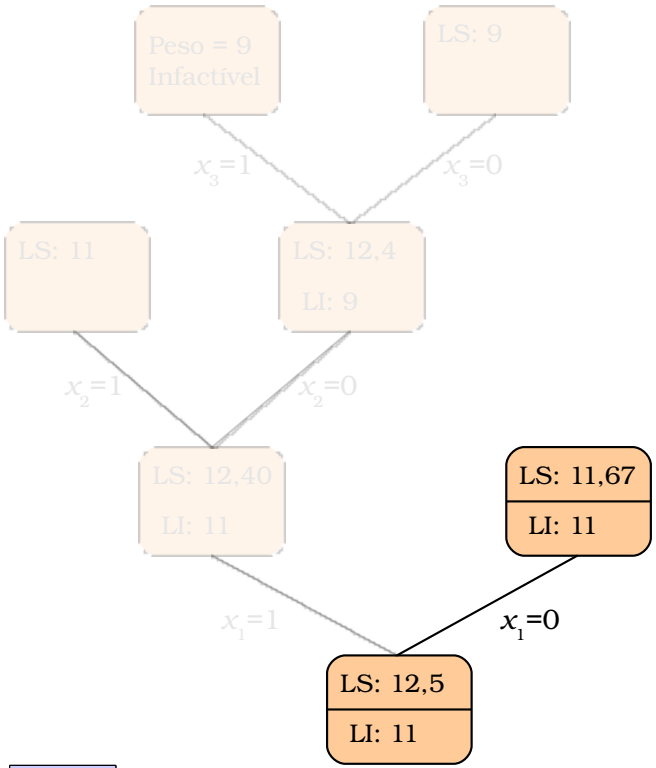
C=7

i	l_i	p_i	l_i/p_i	x_i
1	7	4	1,75	?
2	4	3	1,33	?

i	l_i	p_i	l_i/p_i	x_i
3	9	5	1,80	?
4	2	2	1,00	?

Incumbente: 11
Podado por Limite: como os lucros são inteiros, o melhor lucro possível considerando a relaxação 11,67 é 11.

Branch and Bound - Exemplo: Problema da Mochila



C=7

i	l_i	p_i	l_i/p_i	x_i
1	7	4	1,75	?
2	4	3	1,33	?

i	l_i	p_i	l_i/p_i	x_i
3	9	5	1,80	?
4	2	2	1,00	?

Incumbente: 11

Solução

Provadamente ótima

Universidade Federal de Ouro Preto

Restrições

Restrições Fortes (rígidas)

- Aquelas que obrigatoriamente devem ser satisfeitas.
- Ex.: Enfermeiras não podem trabalhar em dois turnos diferentes em um mesmo dia

Restrições Fracas (flexíveis)

- Conjunto de restrições que devem ser satisfeitas se possível, mas para as quais sabemos que nem todas poderão ser atendidas;
- Ex.: Determinada enfermeira prefere trabalhar nos turnos diurnos;

Restrições

Restrições Fortes (rígidas)

- Aquelas que obrigatoriamente devem ser satisfeitas.
- Ex.: Enfermeiras não podem trabalhar em dois turnos diferentes em um mesmo dia

Restrições Fracas (flexíveis)

- Conjunto de restrições que devem ser satisfeitas se possível, mas para as quais sabemos que nem todas poderão ser atendidas;
- Ex.: Determinada enfermeira prefere trabalhar nos turnos diurnos;

Instâncias

Instâncias

- Disponibilizadas em formato txt e xml;
- Estão classificadas de acordo com a fase a qual pertencem;
- Possuem todas informações referentes aos dados de entrada;

Instâncias

Instâncias

- Disponibilizadas em formato txt e xml;
- Estão classificadas de acordo com a fase a qual pertencem;
- Possuem todas informações referentes aos dados de entrada;

Formulação

Variáveis de decisão

O problema de se encontrar uma solução é formulado em termos das variáveis de decisão, como por exemplo:

$$x_{n,s,d} = \begin{cases} 1, \text{ se a enfermeira } n \text{ foi alocada no turno } s \\ \quad \textit{do dia } d; \\ 0, \text{ caso contrário;} \end{cases}$$

Função Objetivo

Minimizar a ocorrência das violações das variáveis de folga de acordo com as restrições as quais estão sujeitas em cada restrição. Exemplo:

Minimize:

$$\sum_{n \in N} = \overline{vna}(n)$$

Formulação

Restrições Fortes

- Todos os diferentes tipos de turnos devem ser atribuídos às enfermeiras;
- Uma enfermeira pode apenas trabalhar em um turno por dia, isto é, dois turnos diferentes não podem ser atribuídos a uma enfermeira em um mesmo dia;

Exemplo

Não trabalhar mais de um turno diariamente

$$\sum_{s \in S} x_{n,s,d} \leq 1$$

$$\forall n \in N, j \in \{1...nW\}, d \in W_j$$

Formulação

Restrições Fracas

Não necessariamente devem ser satisfeitas e servem para avaliar a qualidade do escalonamento.

Exemplo do modelo

Número máximo de finais de semana consecutivos trabalhando

$$\sum_{i_1 \in \{i \ldots i+\overline{nwec}(c(n))\}} y_{n,i_1} \leq \overline{nwec}(c(n)) + \overline{vnwec}(n)$$

$$\forall n \in N, i \in \{1 \ldots nWe(c(n))\} - \overline{nwec}(c(n)) : \overline{n}wec(cn(n)) <> 0$$

Análise de Complexidade

Complexidade

- O método Branch-and-Bound contém em sua estrutura de dados uma árvore binária.
- usando somente o *Branch* temos um algoritmo exato que em um número finito de passos fornece a solução ótima;
  - Extremamente **ineficiente** !
  - para n variáveis binárias temos 2^n nós a serem explorados.
- O segredo para melhorar a eficiência do BB é a poda de algumas sub-árvores através do uso de limites.

Universidade Federal de Ouro Preto

Análise de Complexidade

Complexidade

- O método Branch-and-Bound contém em sua estrutura de dados uma árvore binária.
- usando somente o *Branch* temos um algoritmo exato que em um número finito de passos fornece a solução ótima;
  - Extremamente **ineficiente** !
  - para n variáveis binárias temos 2^n nós a serem explorados.
- O segredo para melhorar a eficiência do BB é a poda de algumas sub-árvores através do uso de limites.

Universidade Federal de Ouro Preto

Análise de Complexidade

Complexidade

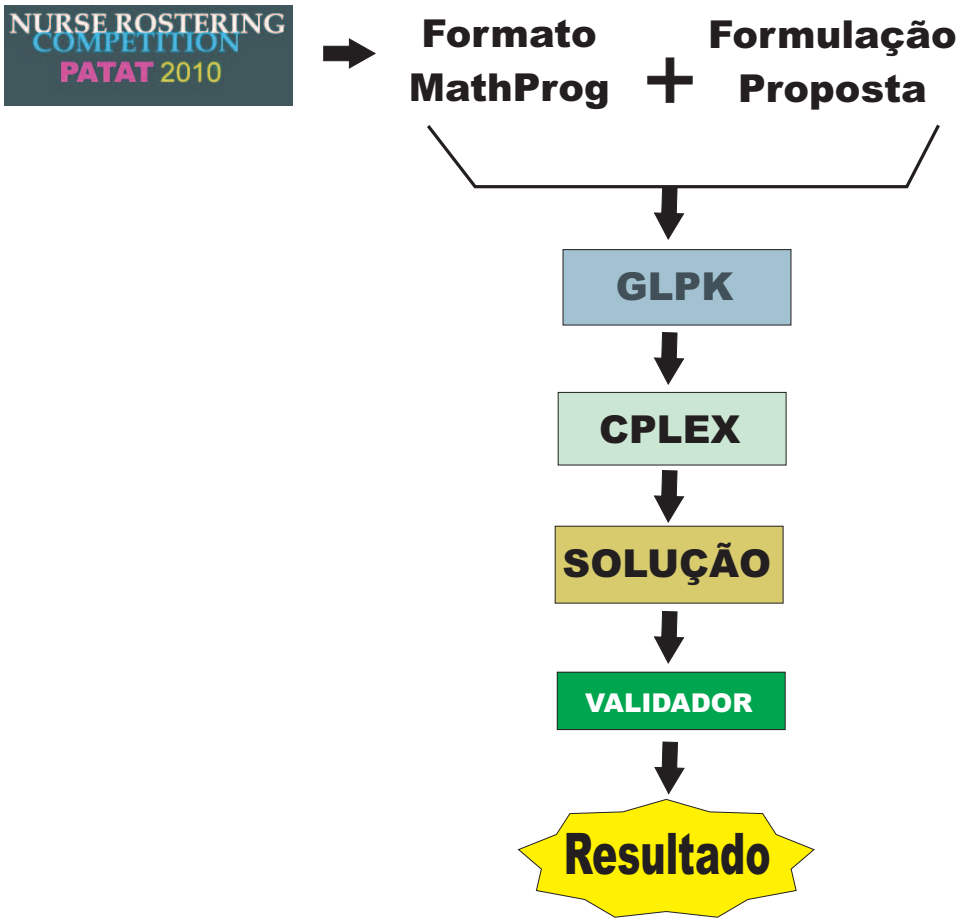
- O método Branch-and-Bound contém em sua estrutura de dados uma árvore binária.
- usando somente o *Branch* temos um algoritmo exato que em um número finito de passos fornece a solução ótima;
  - Extremamente **ineficiente** !
  - para n variáveis binárias temos 2^n nós a serem explorados.
- O segredo para melhorar a eficiência do BB é a poda de algumas sub-árvores através do uso de limites.

Análise de Complexidade

Complexidade

- O método Branch-and-Bound contém em sua estrutura de dados uma árvore binária.
- usando somente o *Branch* temos um algoritmo exato que em um número finito de passos fornece a solução ótima;
  - Extremamente **ineficiente** !
  - para n variáveis binárias temos 2^n nós a serem explorados.
- O segredo para melhorar a eficiência do BB é a poda de algumas sub-árvores através do uso de limites.

Etapas


NURSE ROSTERING
COMPETITION
PATAT 2010
Formato
MathProg
+
Formulação
Proposta
GLPK
CPLEX
SOLUÇÃO
VALIDADOR
Resultado

Máquina utilizada

- Processador Core I5 3.2 GHz;
- 16 GB Ram;
- Sistema Operacional Linux;
- Tempo de execução: Ilimitado.

Resultados

Tabela: Resultados

Instâncias	Penalidade	Competição
sprint_late01	228	37
sprint_late02	139	42
sprint_late06	348	42
sprint_late10	383	43

Conclusão e Trabalhos futuros

Conclusões

- A utilização de métodos exatos produz bons resultados;
- Método *branch-and-Bound* através da poda se torna um método eficiente na prática;

Trabalhos Futuros

- Utilização do método de geração de colunas;
- Utilização de busca local;
- Utilização de heurísticas para geração de soluções iniciais para os métodos exatos.

Universidade Federal de Ouro Preto

Conclusão e Trabalhos futuros

Conclusões

- A utilização de métodos exatos produz bons resultados;
- Método *branch-and-Bound* através da poda se torna um método eficiente na prática;

Trabalhos Futuros

- Utilização do método de geração de colunas;
- Utilização de busca local;
- Utilização de heurísticas para geração de soluções iniciais para os métodos exatos.

Universidade Federal de Ouro Preto

Conclusão e Trabalhos futuros

Conclusões

- A utilização de métodos exatos produz bons resultados;
- Método *branch-and-Bound* através da poda se torna um método eficiente na prática;

Trabalhos Futuros

- Utilização do método de geração de colunas;
- Utilização de busca local;
- Utilização de heurísticas para geração de soluções iniciais para os métodos exatos.

Universidade Federal de Ouro Preto

Conclusão e Trabalhos futuros

Conclusões

- A utilização de métodos exatos produz bons resultados;
- Método *branch-and-Bound* através da poda se torna um método eficiente na prática;

Trabalhos Futuros

- Utilização do método de geração de colunas;
- Utilização de busca local;
- Utilização de heurísticas para geração de soluções iniciais para os métodos exatos.

Universidade Federal de Ouro Preto

Fim

PERGUNTAS?