Pesquisa Operacional Aplicada à Mineração

Módulo de Otimização – Parte III

Prof. Marcone J. F. Souza
Prof. Túlio A. M. Toffolo

marcone.freitas@yahoo.com.br | tulio@toffolo.com.br

Departamento de Computação
Universidade Federal de Ouro Preto

- **Prof. Marcone Jamilson Freitas Souza**
  Departamento de Computação
  Universidade Federal de Ouro Preto
  www.decom.ufop.br/prof/marcone
  marcone.freitas@yahoo.com.br

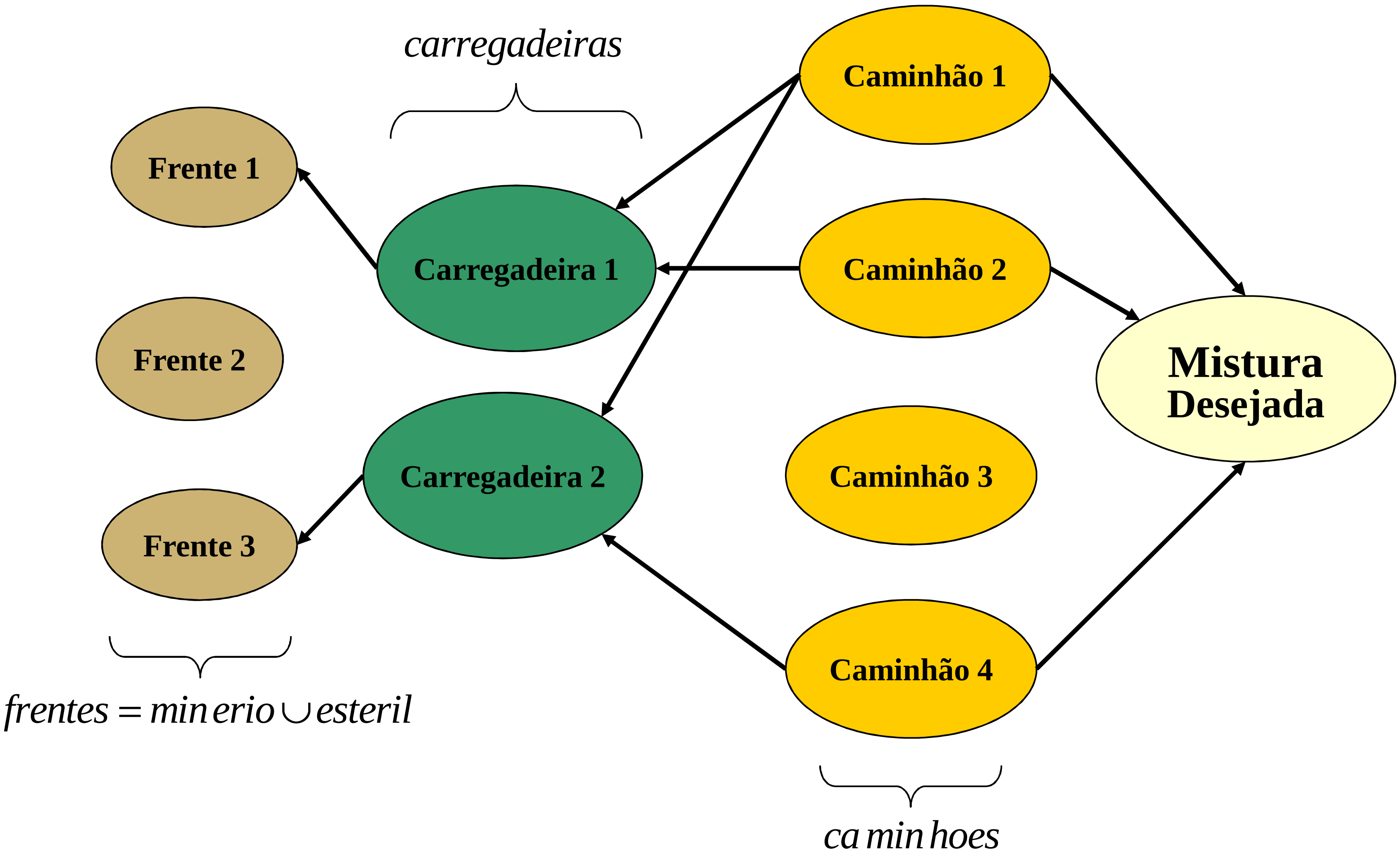
- **Prof. Túlio Ângelo Machado Toffolo**
  Departamento de Computação
  Universidade Federal de Ouro Preto
  www.decom.ufop.br/toffolo
  tulio@toffolo.com.br

Roteiro

- Problema da Alocação Dinâmica de Caminhões
- Aplicações de técnicas de otimização na **Vale**
- Problema da Seleção de Projetos
- Problema do Caixeiro Viajante
- Heurísticas computacionais para otimização
  - Conceitos básicos
  - Heurísticas construtivas
  - Heurísticas clássicas de refinamento
  - Metaheurísticas

PROBLEMA DA ALOCAÇÃO DINÂMICA DE CAMINHÕES

Alocação Dinâmica de Caminhões



Alocação Dinâmica de Caminhões

<u>Dados de entrada (1):</u>

- t_{ij}: Teor do parâmetro j na frente i (%);
- tl_j: Teor mínimo admissível para o parâmetro j (%);
- tu_j: Teor máximo admissível para o parâmetro j (%);
- tr_j: Teor recomendado para o parâmetro j (%);
- wnm_j: Peso por desvio negativo para o parâmetro j;
- wpm_j: Peso por desvio positivo para o parâmetro j;
- wpp: Peso por desvio positivo de produção;
- wnp: Peso por desvio negativo de produção;

Alocação Dinâmica de Caminhões

Dados de entrada (2):

- Qu_i: Massa disponível na frente i (t);
- $tempCiclo_i$: Tempo de ciclo de caminhões para a frente i;
- $estMin_i$: Se a frente i é de minério (1) ou estéril (0);
- Cu_k: Produção máxima da carregadeira k (t/h);
- Cl_k: Produção mínima da carregadeira k (t/h);
- $capCam_l$: Capacidade do caminhão l (t);
- $comp_{lk}$: Se o caminhão l é compatível (1) ou não (0) com a carregadeira k;
- rem: Relação estéril/minério.

Alocação Dinâmica de Caminhões

Variáveis de decisão:

- x_i: Ritmo de lavra para a frente i (t/h);
- y_{ik}: 1 se a carregadeira k opera na frente i e 0 c.c.;
- $usou_l$ = 1 se o caminhão l for usado e 0 caso contrário;
- n_{li}: Viagens que o caminhão l realiza à frente i;
- dnm_j e dpm_j: Desvios negativo e positivo da meta do parâmetro j (t/h);
- dnu_l e dpu_l: Desvios negativo e positivo de utilização do caminhão l;
- dnp e dpp: Desvios negativo e positivo de produção;

Alocação Dinâmica de Caminhões

- Função objetivo

$$\min \sum_{j \in Parametros} \left(wnm_j dnm_j + wpm_j dpm_j \right) +$$
$$wnp \cdot dnp + wnp \cdot dpp + \sum_{l \in Caminhoes} CapCam_l usou_l$$

Alocação Dinâmica de Caminhões

- Admite-se que haja falta (dnm_j) ou excesso (dpm_j) do parâmetro j na mistura em relação à meta de qualidade

$$\sum_{\substack{i \in Frentes \\ |estMin_i = 1}} \left(t_{ij} - tr_j\right) x_i + dnm_j - dpm_j = 0 \quad \forall j \in Parametros$$

- Os desvios dnm_j e dpm_j devem ser penalizados na função objetivo.

Alocação Dinâmica de Caminhões

- Atendimento aos limites de especificação (obrigatório):

$$\sum_{\substack{i \in Frentes \\ | estMin_i = 1}} \left(t_{ij} - tu_j\right) x_i \leq 0 \quad \forall j \in Parametros$$

$$\sum_{\substack{i \in Frentes \\ | estMin_i = 1}} \left(t_{ij} - tl_j\right) x_i \geq 0 \quad \forall j \in Parametros$$

Alocação Dinâmica de Caminhões

- A produção deve respeitar o máximo admitido:

$$\sum_{\substack{i \in Frentes \\ | estMin_i = 1}} x_i \leq pu \quad \forall j \in Parametros$$

- A produção deve respeitar o mínimo admitido:

$$\sum_{\substack{i \in Frentes \\ | estMin_i = 1}} x_i \geq pl \quad \forall j \in Parametros$$

Alocação Dinâmica de Caminhões

- A meta de produção deve ser buscada sempre que possível.

$$\sum_{\substack{i \in Frentes \\ | estMin_i = 1}} x_i + dnp - dpp = pr \quad \forall j \in Parametros$$

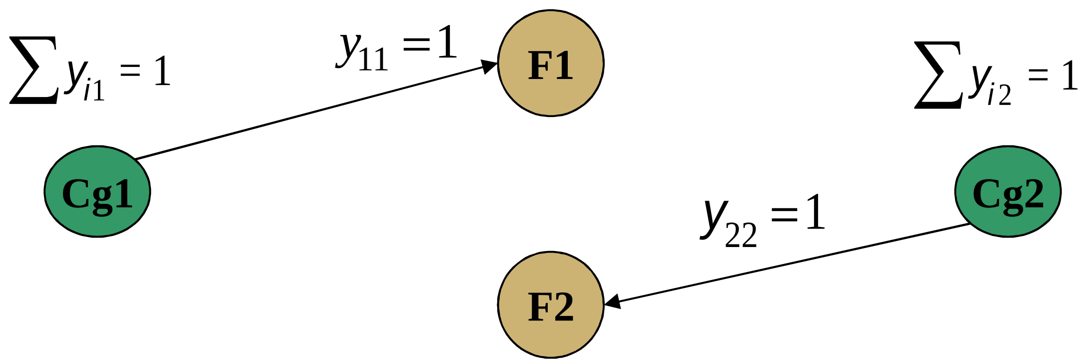
- A relação estéril/minério deve ser atendida:

$$\sum_{\substack{i \in Frentes \\ | estMin_i = 0}} x_i - rem \sum_{\substack{i \in Frentes \\ | estMin_i = 1}} x_i \geq 0 \quad \forall j \in Parametros$$

Alocação Dinâmica de Caminhões

- No máximo uma carregadeira operando em cada frente

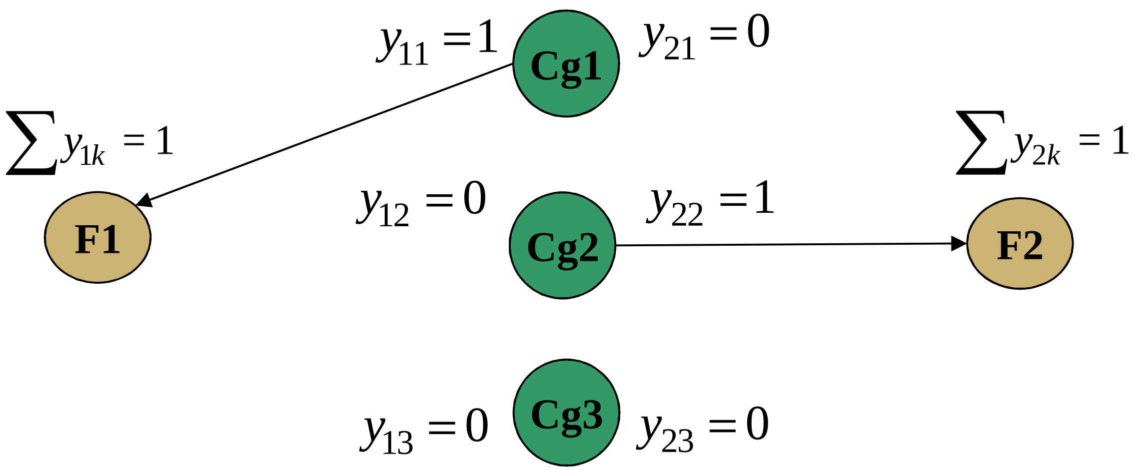
$$\sum_{k \in Carregadeiras} y_{ik} \leq 1 \ \forall i \in Frentes$$



Alocação Dinâmica de Caminhões

- Cada carregadeira deve operar em no máximo uma frente.

$$\sum_{i \in Frentes} y_{ik} \leq 1 \ \forall k \in Carregadeiras$$



Alocação Dinâmica de Caminhões

- O ritmo de lavra da frente i deve ser maior do que a produtividade mínima da carregadeira k alocada à frente

$$x_i \geq \sum_{k \in Carregadeiras} Cl_k y_{ik} \quad \forall i \in Frentes$$

- O ritmo de lavra da frente i deve ser menor do que a produtividade máxima da carregadeira k alocada à frente

$$x_i \leq \sum_{k \in Carregadeiras} Cu_k y_{ik} \quad \forall i \in Frentes$$

Alocação Dinâmica de Caminhões

- Cada caminhão l deve realizar viagens apenas à uma frente i que esteja alocada uma carregadeira compatível

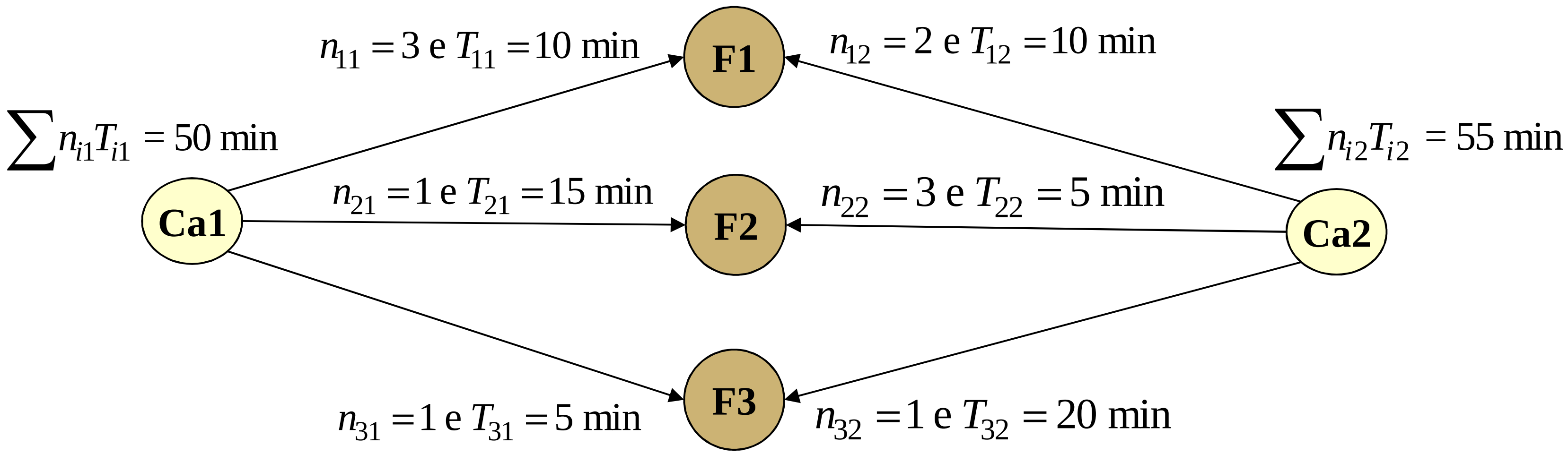
$$n_{il}\, tempCiclo_i \leq \sum_{\substack{k \in Carregadeiras \\ | comp_{lk}=1}} 60 y_{ik} \quad \forall i \in Frentes, \forall l \in Caminhoes$$

$$n_{il} \in \mathbb{Z}^+ \quad \forall i \in Frentes, \forall l \in Caminhoes$$

Alocação Dinâmica de Caminhões

- Cada caminhão l deve operar no máximo 60 minutos multiplicado pela taxa máxima de utilização (tipicamente 80%)

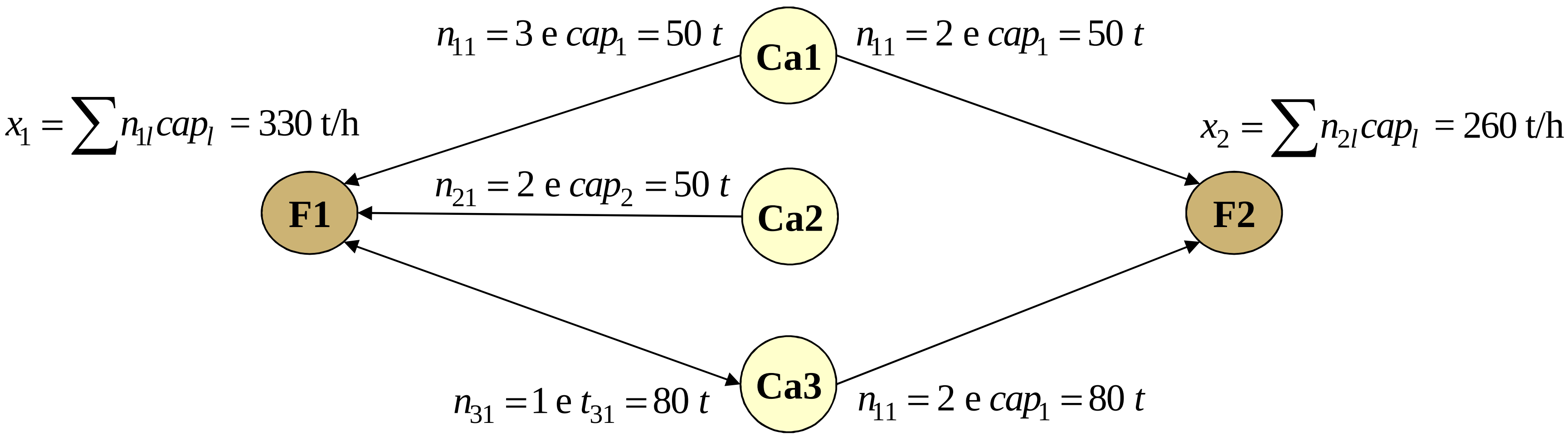
$$\sum_{i \in Frentes} n_{il}\, tempCiclo_i \leq 60\, txMax \quad \forall l \in Caminhoes$$



Alocação Dinâmica de Caminhões

- O ritmo de lavra da frente i deve ser igual à produção realizada pelos caminhões alocados à frente

$$x_i = \sum_{l \in Caminhoes} n_{il}\, capCam_l \quad \forall i \in Frentes$$



Alocação Dinâmica de Caminhões

- Um caminhão é usado se ele faz alguma viagem a alguma frente

$$usou_l \geq \frac{\sum_{i \in Frentes} tempCiclo_i \, n_{il}}{60} \quad \forall l \in Caminhoes$$

$$usou_l \in \{0,1\} \quad \forall l \in Caminhoes$$

APLICAÇÃO DE TÉCNICAS DE OTIMIZAÇÃO NA VALE

Aplicações

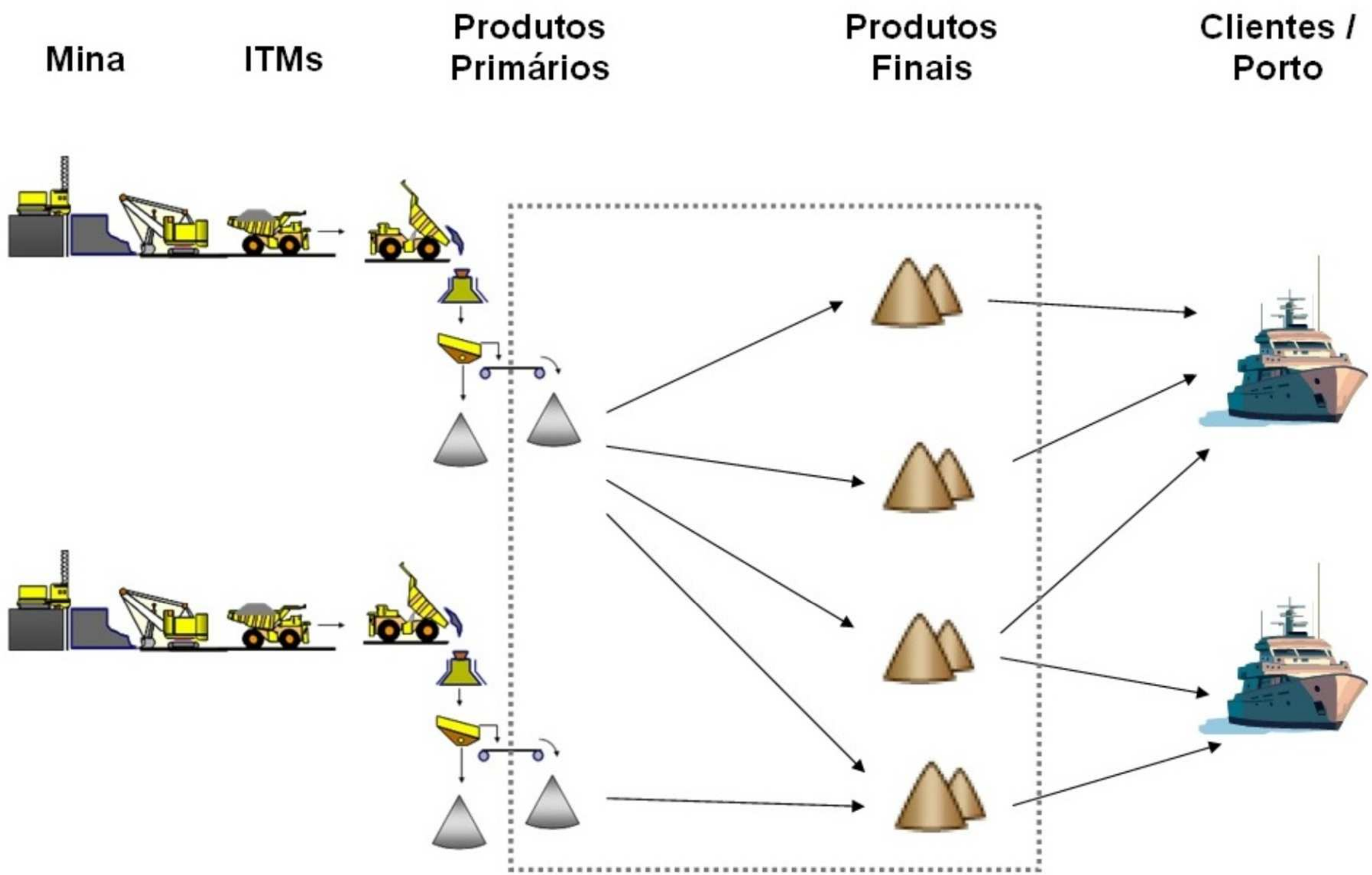
Principais sistemas de **Otimização** desenvolvidos por nós que são utilizados pela VALE.

- **COMPÕE**
  - **Otimização do Planejamento do Fluxo de Produtos**
- **OPTITRENS**
  - **Otimização do Planejamento de Carregamento de Trens**
- **OPTIPILHAS**
  - **Otimização do Planejamento Diário de Lavra**

COMPÕE

OTIMIZAÇÃO DO PLANEJAMENTO DO FLUXO DE PRODUTO

Processo Produtivo de uma Mineradora



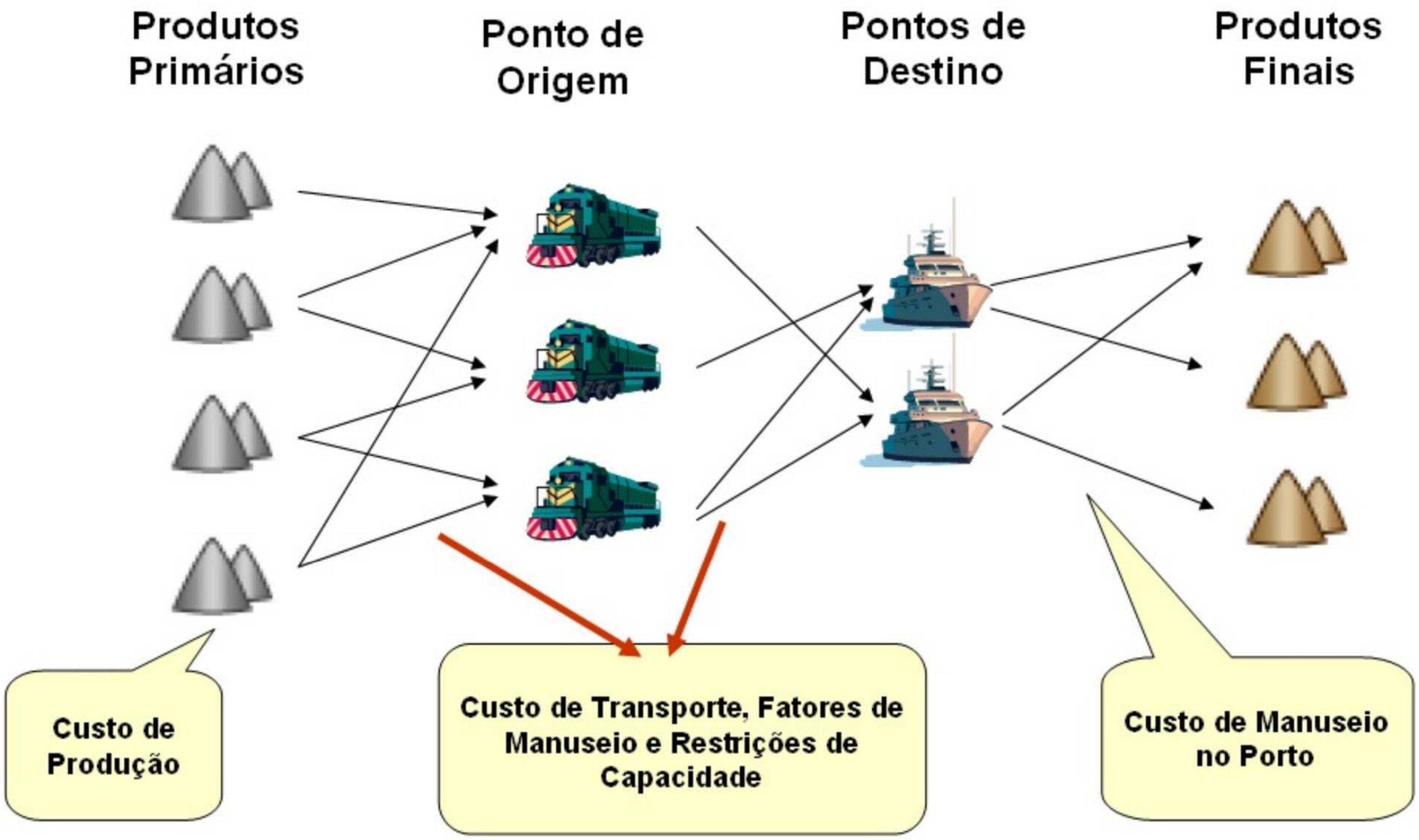
Objetivos

- Atendimento às demandas
- Atendimento aos requisitos de qualidade
- Minimização dos custos com transporte

Motivações

- Problema real com aplicação prática imediata
- O problema abordado integra problemas clássicos da literatura
  - Mistura de minérios
  - Programação e sequenciamento da produção
- Desenvolvimento de uma ferramenta computacional para apoio à tomada de decisão

Descrição do Problema



Produtos Primários

Estoque	Fator de Manuseio "Mina-trem"							
	Fe	SiO_2	Al_2O_3	P	Mn	H_20	OS	…

Para cada trimestre:

Produção	Teores da Produção Trimestral							
	Fe	SiO_2	Al_2O_3	P	Mn	H_20	OS	…

Terminais e Modais de Transporte (local de produção)

Possibilidades de escoamento do minério

Produtos Finais

Especificação (Metas)							
Fe	SiO_2	Al_2O_3	P	Mn	H_20	OS	…

Demanda			
1º	2º	3º	4º

Pesos na Função Objetivo								
Demanda	Fe	SiO_2	Al_2O_3	P	Mn	H_20	OS	…

Desvios e Fator de Manuseio							
Fe	SiO_2	Al_2O_3	P	Mn	H_20	OS	…

Tipo de Transporte (ponto de entrega):
Ferroviário, rodoviário ou via dutos?

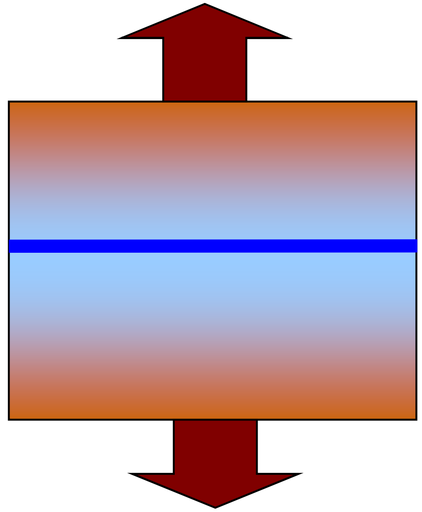
Especificação Meta

- Caso em que a meta e os limites são especificados:

Desvio do Limite Superior

DESVIO DA META

Desvio do Limite Inferior



Exemplo de Composição

Origem	N.º Trens	Qtde (kt)	Fe	SiO_2	Al_2O_3	P	Mn	H_2O	+6.3	+1	-0
TF-01	**170**	**2.210**	**66,78**	**1,79**	**0,80**	**0,054**	**0,205**	**10,1**	**9,5**	**38,2**	**3**
F-05 (Un. 03, ITM-SEC-2)	-	1.183	67,43	0,98	0,52	0,055	0,069	8,5	11,0	21,0	5
-06 (Un. 03, ITM-UMD-1)	-	551	67,94	0,83	0,34	0,041	0,415	12,0	7,5	58,0	1
-07 (Un. 03, ITM-UMD-2)	-	476	63,82	4,90	2,01	0,067	0,299	12,0	8,0	58,0	1
TF-05	**245**	**3.185**	**63,77**	**5,72**	**1,34**	**0,068**	**0,120**	**6,8**	**8,0**	**56,1**	**2**
SF-10 (Un. 06, ITM-A)	-	1.843	63,43	6,04	1,55	0,069	0,094	6,4	5,4	58,0	1
SF-11 (Un. 06, ITM-B)	-	865	63,26	6,55	1,19	0,071	0,174	7,5	14,5	48,9	2
SF-12 (Un. 06, ITM-D)	-	476	66,04	3,00	0,83	0,062	0,125	7,0	6,5	61,5	2
TF-06	**77**	**1.001**	**65,41**	**1,84**	**1,77**	**0,033**	**0,398**	**8,1**	**14,1**	**50,0**	**2**
SF-05 (Un. 05, ITM-Aux)	-	525	65,48	1,84	1,92	0,041	0,415	8,1	17,2	50,0	2
SF-06 (Un. 05, ITM-Sec)	-	476	65,34	1,84	1,61	0,024	0,379	8,0	10,7	50,0	3
TF-08	**104**	**1.352**	**64,36**	**4,88**	**1,22**	**0,059**	**0,020**	**9,3**	**2,6**	**49,0**	**2**
-18 (Compras, Compras)	-	1.352	64,36	4,88	1,22	0,059	0,020	9,3	2,6	49,0	2
TF-09	**625**	**8.125**	**65,85**	**2,48**	**1,44**	**0,058**	**0,051**	**7,6**	**18,0**	**53,0**	**2**
01 (Un. 01, ITM-A) - Estoque	-	1.500	64,36	3,47	1,67	0,068	0,014	7,9	21,4	48,0	2
CSF-01 (Un. 01, ITM-A)	-	2.595	65,41	2,85	1,75	0,075	0,022	8,2	23,4	48,0	2
CSF-02 (Un. 01, ITM-M)	-	832	65,41	2,85	1,75	0,075	0,022	8,2	23,4	48,0	2
SF-13 (Un. 07, ITM-A)	-	3.198	67,02	1,61	1,00	0,035	0,100	6,9	10,7	60,6	1
Produto Final Obtido	**1.221**	**15.872**	**65,41**	**3,20**	**1,33**	**0,058**	**0,106**	**8,0**	**13,3**	**51,0**	**2**

Limite Superior Admitido			4,18	1,98	0,072	0,440	8,5	16,0	55,0	3
Demanda do Produto F-SF-09 (D-02)	**15.872**	65,00	3,20	1,20	0,049	0,250	6,5	10,0	51,0	2
Limite Inferior Admitido		64,40					4,5	4,0	47,0	1

Funcionalidades / Restrições

- *Definir blendagens* - Quais produtos iniciais podem compor determinado produto final?
- *Impor blendagens* - Permite ao usuário impor uma blendagem.
- *Atingir meta* – Permite ao usuário determinar o valor exato de uma propriedade na mistura de um determinado produto final.
- *Utilizar produto* – Permite ao usuário a forçar a utilização de uma determinada quantidade de um produto primário.
- etc...

Formulação Matemática

- Funções objetivo consideradas:

$$
\min \begin{cases}
F_1 = \sum_{j \in F} \beta_j^- dp_j^- \\
F_2 = \sum_{j \in F} \sum_{k \in S} \alpha_{jk}^- \omega_{jk} dt_{jk}^- + \sum_{j \in F} \sum_{k \in S} \alpha_{jk}^+ \omega_{jk} dt_{jk}^+ \\
F_3 = \sum_{(i,j) \in B} \sum_{r \in R^B(i,j)} \sum_{t = pt_j^L}^{pt_j^U} cs_t^r x_{ijt}^r + \sum_{c \in C} \sum_{t \in T} cc_{ct} z_{ct}^C
\end{cases}
$$

- Hierarquia bem definida entre os objetivos.
  - Ordem de prioridade considerada: F_1, F_2 e F_3.

Formulação Matemática

Restrições de demanda mínima a ser atingida

$$\sum_{i|(i,j)\in B} \sum_{r\in R^B(i,j)} \sum_{t=pt_j^L}^{pt_j^U} x_{ijt}^r \geq pl_j \qquad \forall j \in F$$

Restrições de atendimento às demandas

$$\sum_{i|(i,j)\in B} \sum_{r\in R^B(i,j)} \sum_{t=pt_j^L}^{pt_j^U} x_{ijt}^r + dp_j^- = pr_j \qquad \forall j \in F$$

Formulação Matemática

Restrições de meta de qualidade

$$\sum_{i|(i,j)\in B}\sum_{n\in C^N}\sum_{r\in R^C(i,j,n)}\sum_{t=pt_j^L}^{pt_j^U}(q_{ikt}-tr_{jk})x_{ijt}^r + \sum_{i|(i,j)\in B}\sum_{f\in C^F}\sum_{r\in R^C(i,j,f)}\sum_{t=pt_j^L}^{pt_j^U}(q_{ikt}+fm_{ik}-tr_{jk})x_{ijt}^r + dt_{jk}^- - dt_{jk}^+ = 0 \qquad \forall j\in F, \forall k \in S, \exists\; tr_{jk}$$

Restrições de disponibilidade dos produtos primários

$$\sum_{j|(i,j)\in B}\sum_{r\in R^B(i,j)} x_{ijt}^r \leq o_{it} \qquad \forall i\in P, \forall t\in T$$

Formulação Matemática

. . .

Formulação completa disponível em:

- http://www.decom.ufop.br/toffolo

Resolução do Modelo Multiobjetivo

Metodologia utilizada: **Problemas ϵ-Restrito**

- Apenas uma função objetivo é otimizada por vez.
- Problema multiobjetivo é convertido em n problemas mono-objetivo, sendo n o número de funções objetivo.
- Para cada função objetivo, um valor ϵ_i, $i = 1, ..., n - 1$ é associado a uma função objetivo (ex: $F_1 \leq \epsilon_1$).
- Ao otimizar a função objetivo j, as restrições $F_i = \epsilon_i, \forall i \in 1, ..., j - 1$ são geradas.

Resolução do Modelo Multiobjetivo

1ª Iteração) Resolução do Problema Considerando F_1

- Apenas a função objetivo F_1 é otimizada.
- As restrições relaxadas de qualidade não são geradas.
- $\epsilon_1 \leftarrow$ valor obtido para F_1.

2ª Iteração) Resolução do Problema Considerando F_2

- Apenas a função objetivo F_2 é otimizada.
- Todas as restrições do modelo são geradas, além de:

$$F_1 = \sum_{j \in F} \beta_j^- dp_j^- = \epsilon_1$$

- $\epsilon_2 \leftarrow$ valor obtido para F_2.

Resolução do Modelo Multiobjetivo

3ª Iteração) Resolução do Problema Considerando F_3

- Apenas a função objetivo F_3 é otimizada.
- Todas as restrições do modelo são geradas, além de:

$$F_1 = \sum_{j \in F} \beta_j^- dp_j^- = \epsilon_1$$

$$F_2 = \sum_{j \in F} \sum_{k \in S} \alpha_{jk}^- \omega_{jk} dt_{jk}^- + \sum_{j \in F} \sum_{k \in S} \alpha_{jk}^+ \omega_{jk} dt_{jk}^+ = \epsilon_2$$

- Desvantagens:
  - Resolver 3 problemas
  - Pode acarretar um aumento na complexidade, ao inserir até $n - 1$ restrições adicionais.

Testes

3 instâncias geradas para cada cenário

- Inst. 1: demanda pouco menor do que a produção
- Inst. 2: demanda equivale a cerca de 50% da produção
- Inst. 3: demanda equivale a cerca de 150% da produção

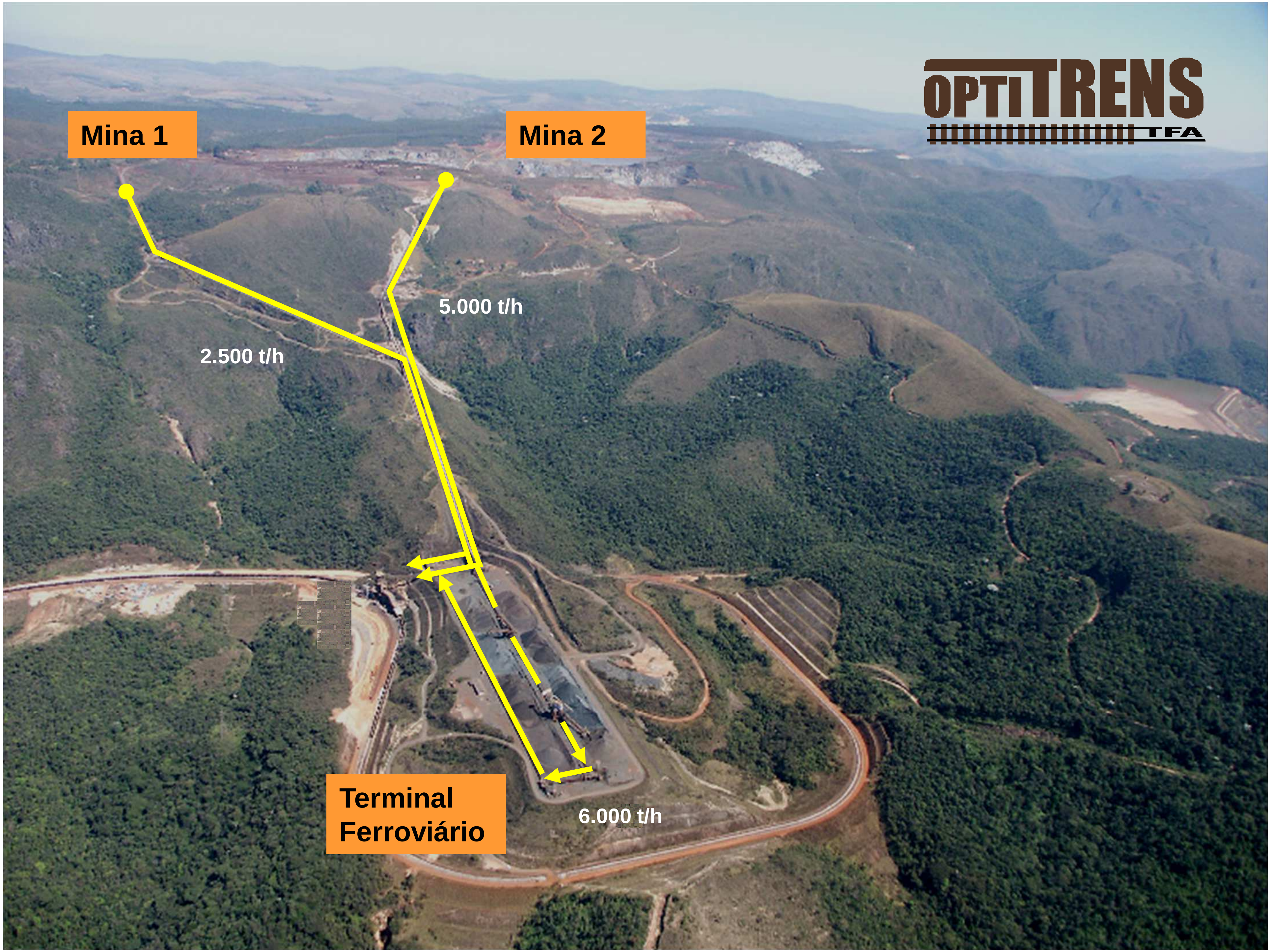
- **Cenário anual**: apenas um período de tempo, $|T| = 1$
- **Cenário trimestral e mensal**: produção e demandas por períodos, totalizando um ano ($|T| = 4$ e $|T| = 12$)
- **Cenário diário**: demandas possuem data limite de atendimento ($|T| = 15$ e $|T| = 30$)

Conclusões

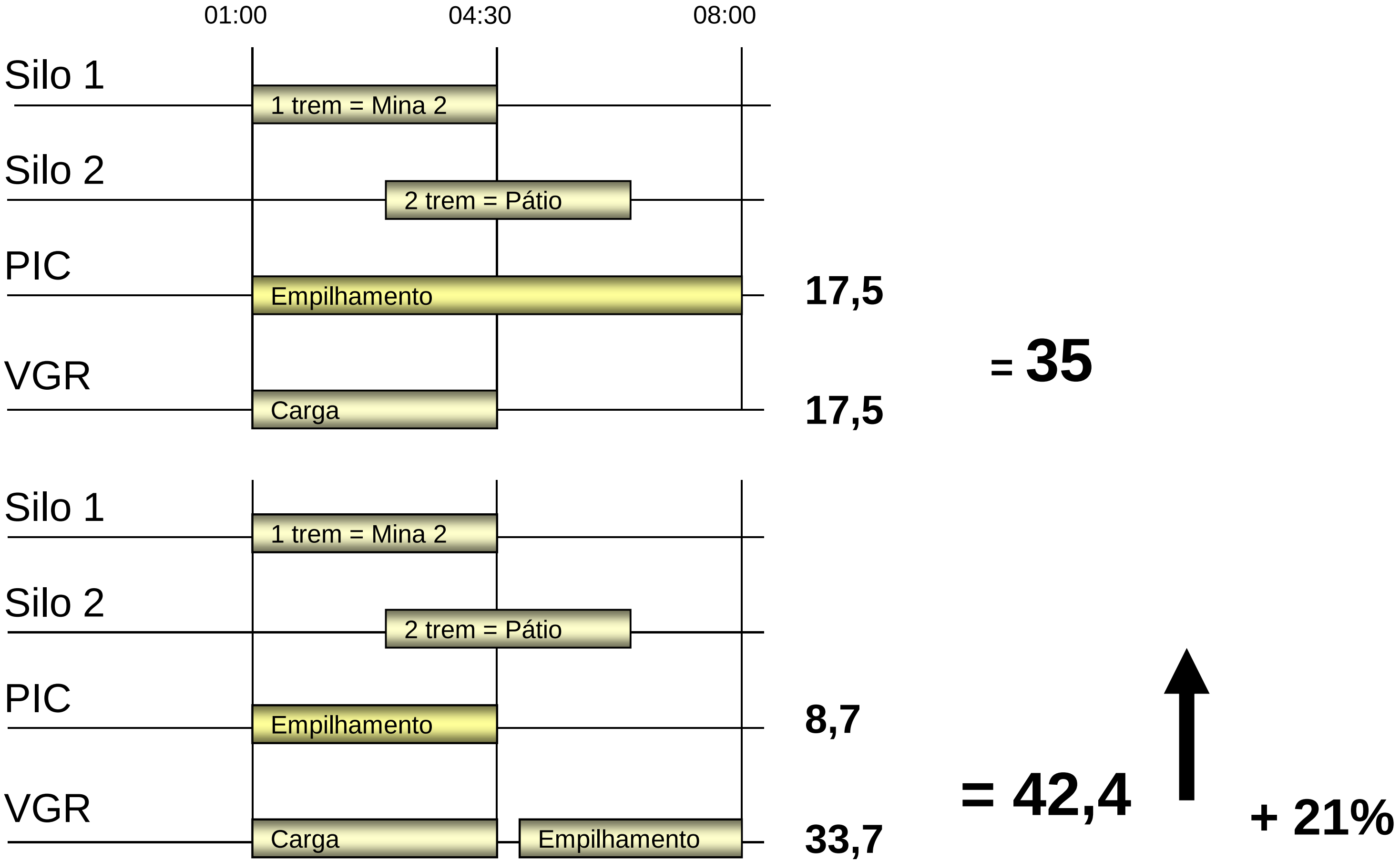
- Diminuição drástica do tempo de tomada de decisão
  - Redução: **cerca de 7 dias para alguns segundos**;
- Possibilita a analise de diversos cenários para a tomada de decisão
- Solução gerada é **ótima** !!!

OPTITRENS

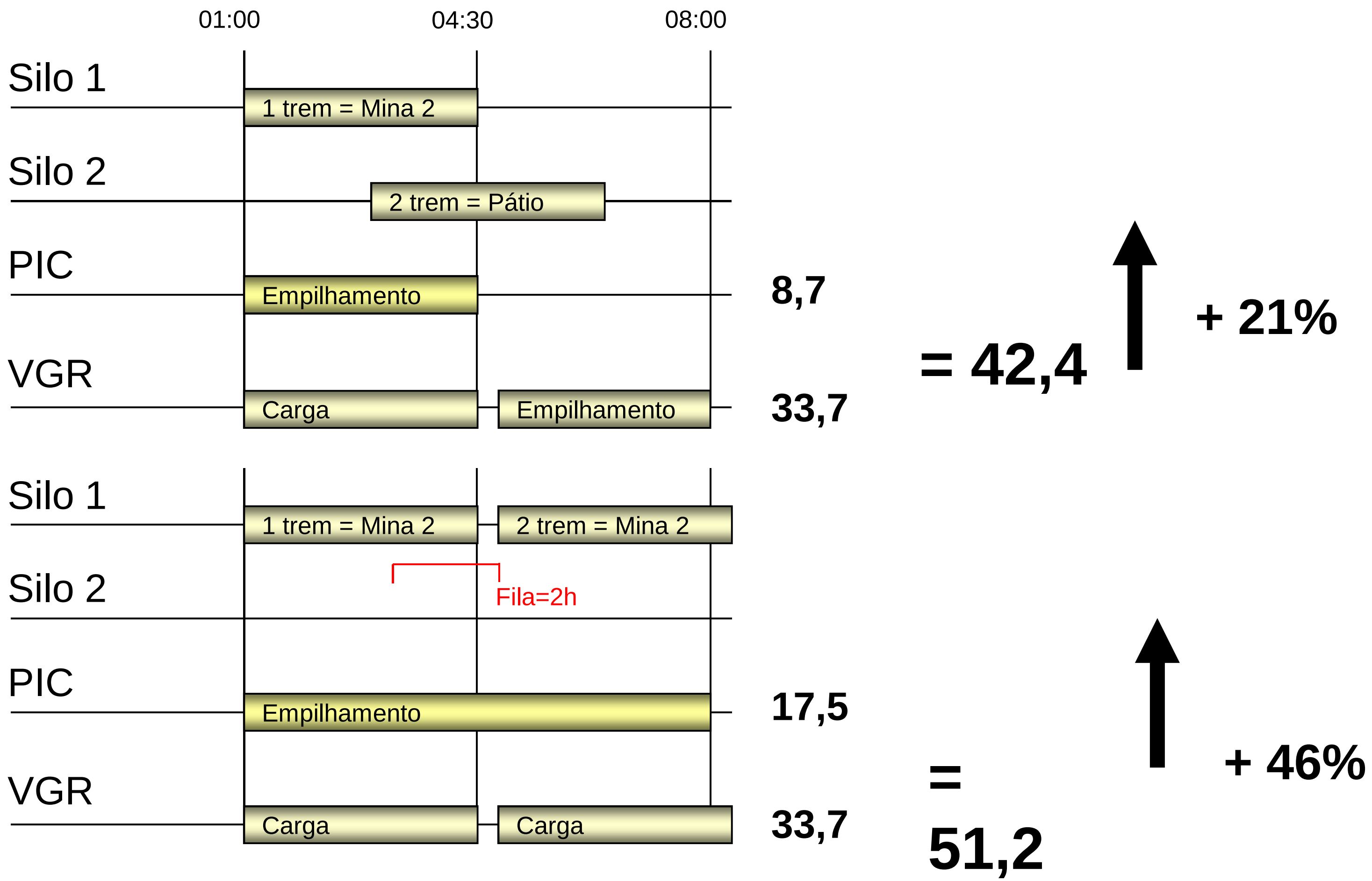
OTIMIZAÇÃO DO PLANO
DE CARREGAMENTO DE TRENS


OPTITRENS
TFA
Mina 1
Mina 2
5.000 t/h
2.500 t/h
Terminal Ferroviário
6.000 t/h

Exemplo de Otimização



Exemplo de Otimização



Objetivos

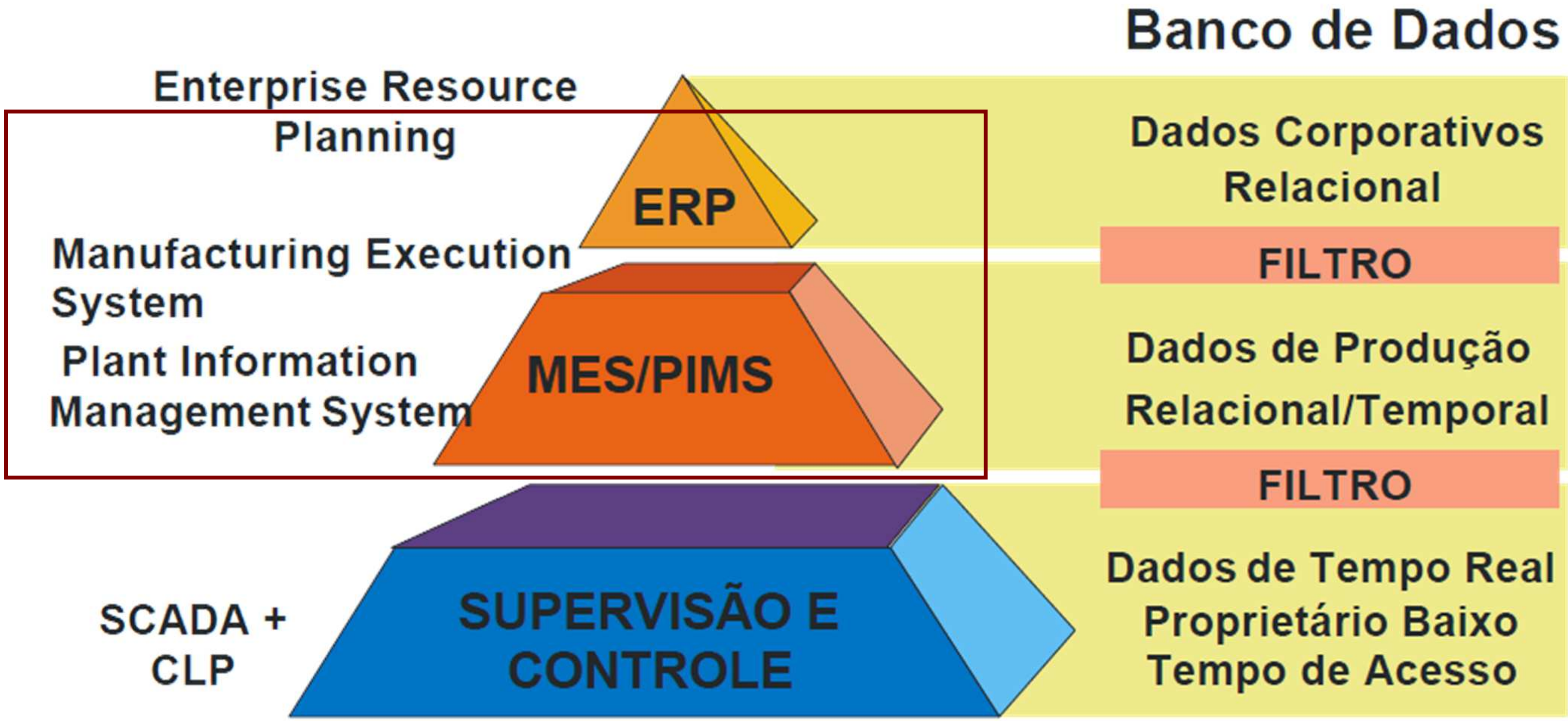
- **Otimizar a logística do terminal**
- Capturar dados do PIMS
- Detectar gargalos na cadeia produtiva
- Armazenar dados históricos de utilização

Algoritmos Utilizados na Otimização

- **Iterated Local Search**

- Principais estrutura de vizinhança:
  - Troca de horário de trem
  - Troca de origem (mina/pátio)
  - Troca de destino (silo)

Utilização do Sistema



Principais Benefícios

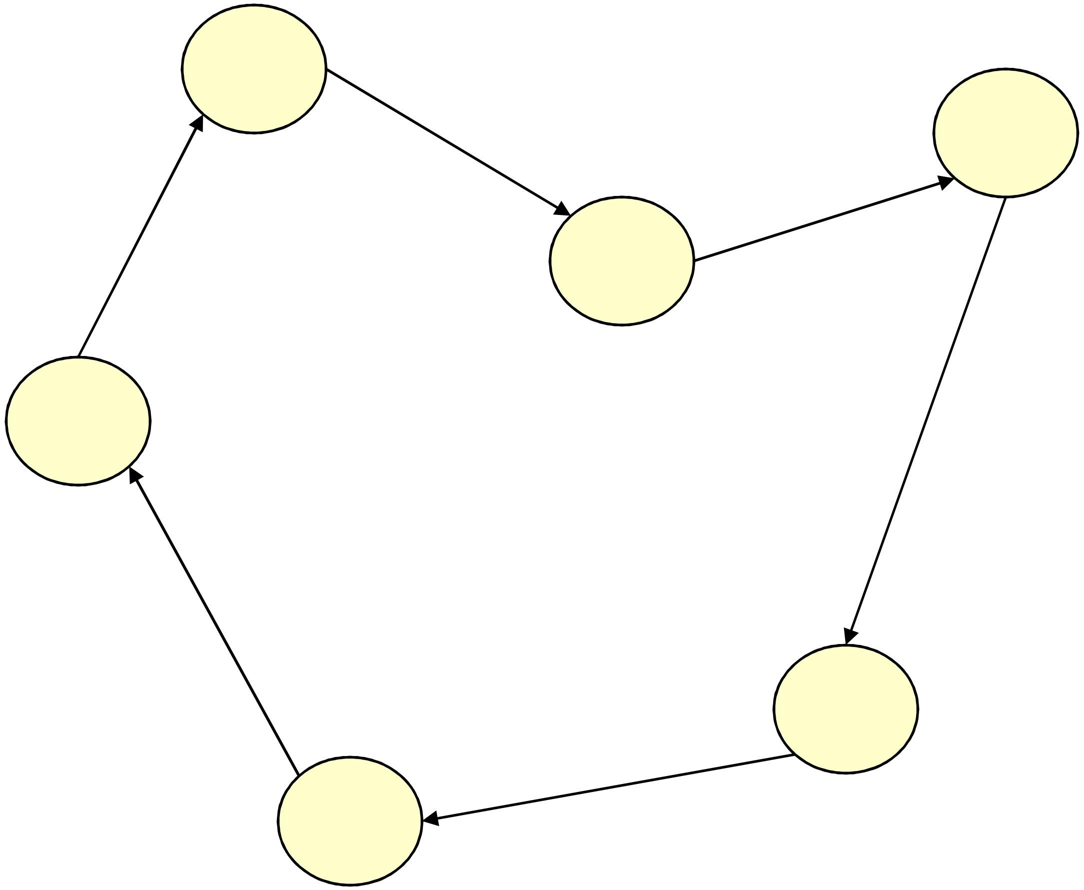
- Melhoria da produtividade do terminal
- Homegeneização das produções dos diferentes turnos
- Fácil identificação de gargalos na cadeia produtiva
- Simplificação da atividade de planejamento e controle de qualidade
- Simplificação da atividade de apontamento

PROBLEMA DO CAIXEIRO VIAJANTE

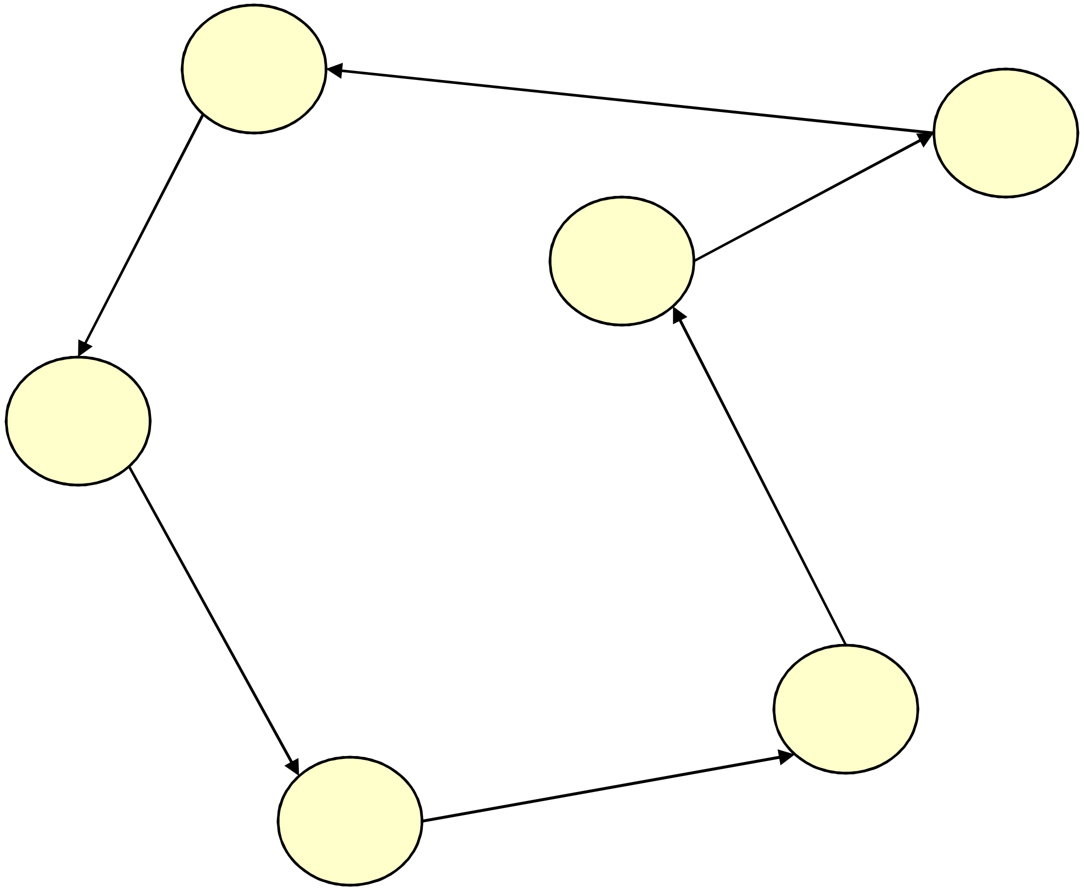
Problema do Caixeiro Viajante

- Dado um conjunto de cidades e uma matriz de distâncias entre elas
- PCV consiste em encontrar uma rota para um Caixeiro Viajante tal que este:
  - parta de uma cidade origem
  - passe por todas as demais cidades uma única vez
  - retorne à cidade origem ao final do percurso
  - percorra a menor distância possível
- Rota conhecida como <u>Ciclo Hamiltoniano</u>

Problema do Caixeiro Viajante



Problema do Caixeiro Viajante



Problema do Caixeiro Viajante

Dados de entrada:

- Cidades: Conjunto de cidades
- d_{ij} = distância entre as cidades i e j

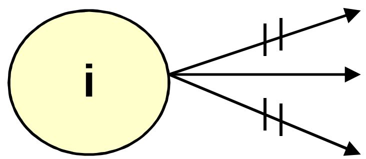
Variáveis de decisão:

- x_{ij} = 1 se a aresta (i,j) será usada; 0, caso contrário
- f_{ij} = quantidade de fluxo de i para j
- Função objetivo:

Problema do Caixeiro Viajante

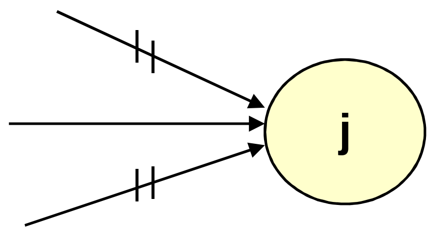
- De cada cidade i só sai uma aresta:

$$\sum_{j \in Cidades} x_{ij} = 1 \quad \forall i \in Cidades$$



- A cada cidade j só chega uma aresta:

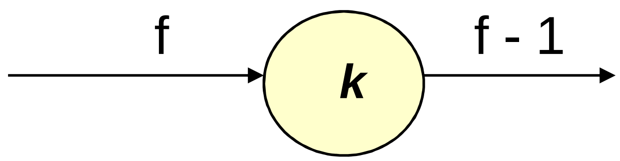
$$\sum_{i \in Cidades} x_{ij} = 1 \quad \forall j \in Cidades$$



Problema do Caixeiro Viajante

- O fluxo que chega a uma cidade i menos o que sai é igual a uma unidade:

$$\sum_{i \in Cidades} f_{ik} - \sum_{j \in Cidades} f_{kj} = 1 \quad \forall k \in Cidades \mid k \neq 1$$



Problema do Caixeiro Viajante

- O fluxo máximo em cada aresta é igual a n-1, onde n é o número de cidades:

$$f_{ij} \leq (n-1)x_{ij} \quad \forall i \in Cidades, \forall j \in Cidades$$

$$f_{ij} \geq 0 \quad \forall i \in Cidades, \forall j \in Cidades$$

$$x_{ij} \in \{0,1\} \quad \forall i \in Cidades, \forall j \in Cidades$$

Problema do Caixeiro Viajante

- Considerando PCV simétrico ($d_{ij} = d_{ji}$), para *n* cidades há (*n*-1)!/2 rotas possíveis
- Para *n* = 20 cidades, há 10^{16} rotas possíveis. Um computador que avalia uma rota em 10^{-8} segundos gastaria cerca de 19 anos para encontrar a melhor solução por enumeração completa
- Métodos de enumeração implícita, tais como *branch-and-bound*, embutidos em *solvers*, podem reduzir significativamente este tempo
- Não há garantia de que isso sempre ocorra

Problema do Caixeiro Viajante

- Para dimensões mais elevadas, a resolução por modelos de programação matemática é proibitiva em termos de tempos computacionais
- PCV é da classe NP-difícil: ainda não existem algoritmos exatos que o resolvam em tempo polinomial
- À medida que *n* cresce, o tempo de execução cresce exponencialmente

Problema do Caixeiro Viajante

- PCV é resolvido por meio de heurísticas:
  - Procedimentos que seguem uma intuição para resolver o problema (forma humana de resolver o problema, fenômenos naturais, processos biológicos, etc.)
  - Não garantem a otimalidade da solução final
  - Em geral, produzem soluções finais de boa qualidade rapidamente

HEURÍSTICAS

PROBLEMA DO CAIXEIRO VIAJANTE

Heurísticas

- Construtivas
  - Consistem em construir uma solução passo a passo, elemento por elemento
- De refinamento
  - Consistem em efetuar modificações na solução construída, de forma a tentar melhorá-la

Heurística de construção gulosa

- Funcionamento:
  - Constrói uma solução, elemento por elemento
  - A cada passo é adicionado um único elemento candidato
  - O candidato escolhido é o “melhor” segundo um certo critério
  - O método se encerra quando todos os elementos candidatos foram analisados

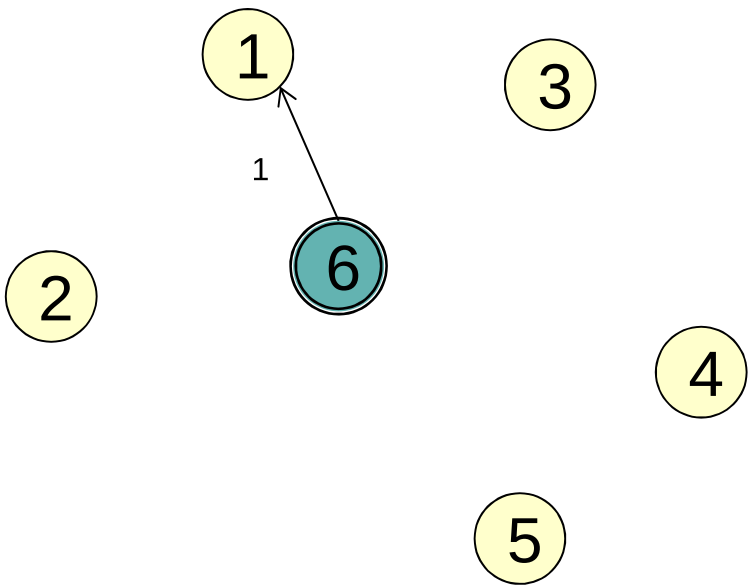
Heurísticas construtivas

- Vizinho mais próximo
  - Idéia central: Construir uma rota passo a passo, adicionando à solução corrente a cidade mais próxima (ainda não visitada) da última cidade inserida
- Inserção mais barata
  - Idéia central: Construir uma rota passo a passo, partindo de rota inicial envolvendo 3 cidades e adicionar a cada passo, a cidade k (ainda não visitada) entre a ligação (i, j) de cidades já visitadas, cujo custo de inserção seja o mais barato

PCV – Vizinho mais Próximo :: Exemplo - Passo 1

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	2
4	4	9	3	0	2	3
5	9	7	8	2	0	2
6	1	2	2	3	2	0

i	j	d_{ij}
6	**1**	1
6	**2**	2
6	**3**	2
6	**4**	3
6	**5**	2

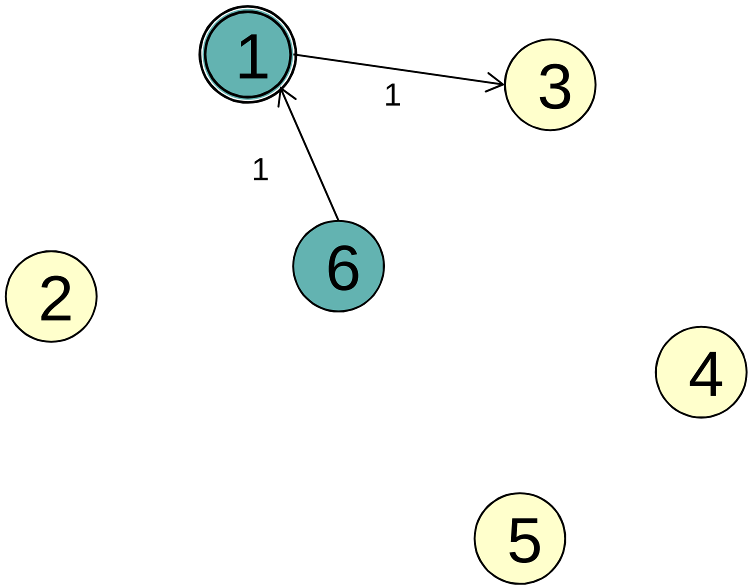


1

PCV – Vizinho mais Próximo :: Exemplo - Passo 2

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	2
4	4	9	3	0	2	3
5	9	7	8	2	0	2
6	1	2	2	3	2	0

i	j	d_{ij}
1	**2**	2
1	**3**	1
1	**4**	4
1	**5**	9

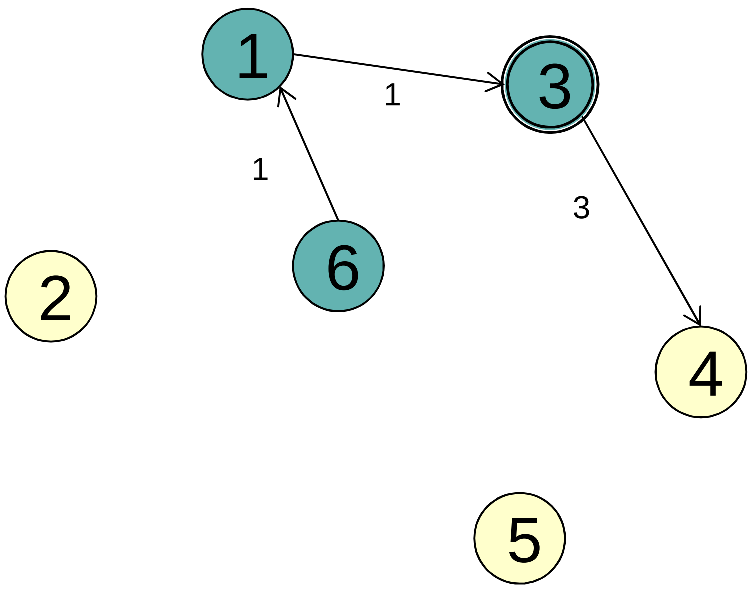


1 + 1 = 2

PCV – Vizinho mais Próximo :: Exemplo - Passo 3

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	2
4	4	9	3	0	2	3
5	9	7	8	2	0	2
6	1	2	2	3	2	0

i	j	d_{ij}
3	**2**	5
3	**4**	3
3	**5**	8

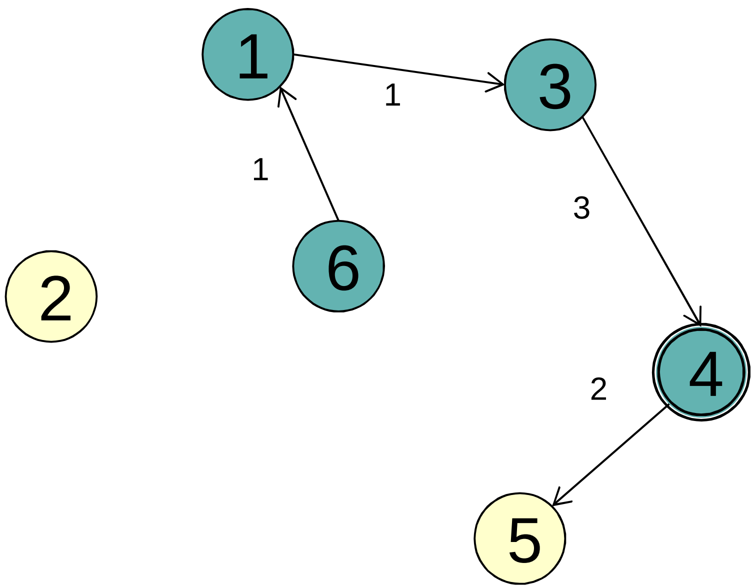


2 + 3 = 5

PCV – Vizinho mais Próximo :: Exemplo - Passo 4

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	2
4	4	9	3	0	2	3
5	9	7	8	2	0	2
6	1	2	2	3	2	0

i	j	d_{ij}
4	**2**	9
4	**5**	2

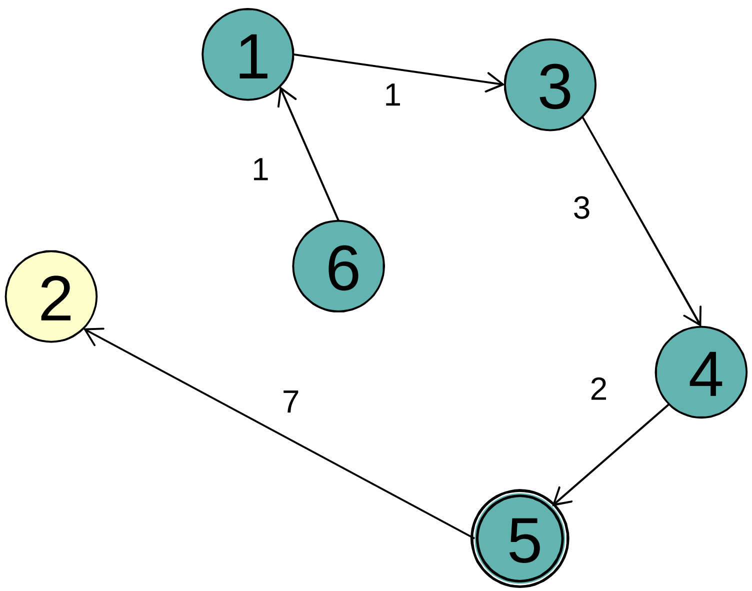


5 + 2 = 7

PCV – Vizinho mais Próximo :: Exemplo - Passo 5

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	2
4	4	9	3	0	2	3
5	9	7	8	2	0	2
6	1	2	2	3	2	0

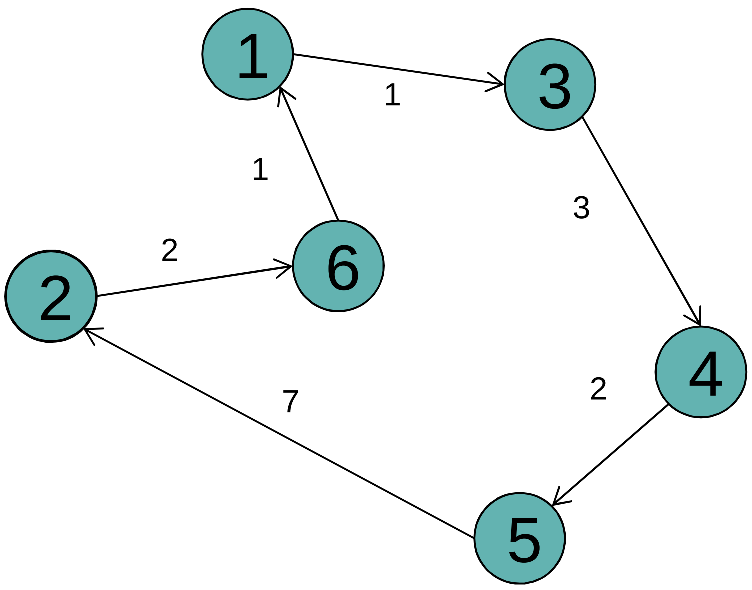
i	j	d_{ij}
5	**2**	7



7 + 7 = 14

PCV – Vizinho mais Próximo :: Exemplo - Passo final

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	2
4	4	9	3	0	2	3
5	9	7	8	2	0	2
6	1	2	2	3	2	0



14 + 2 = 16

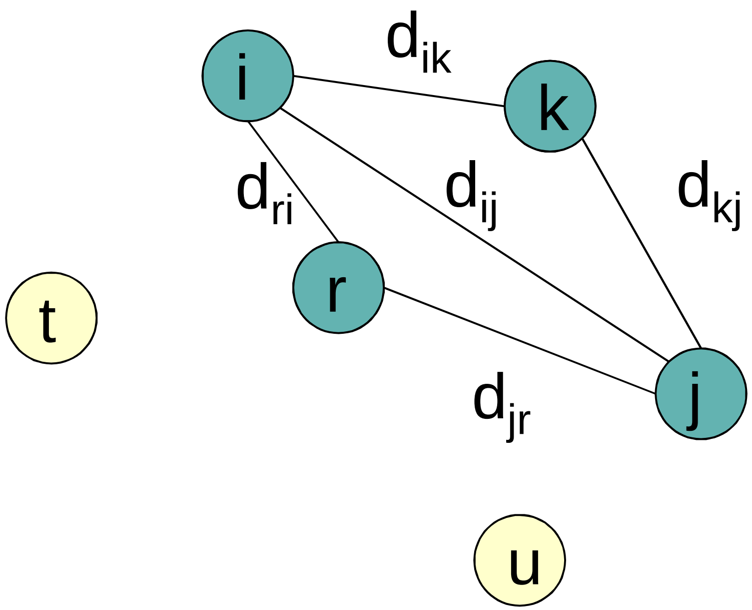
S = (6,1,3,4,5,2)

Heurística da Inserção Mais Barata

- Inserção mais barata
  - Idéia central: Construir uma rota passo a passo, partindo de rota inicial envolvendo 3 cidades (obtidas por um método qualquer) e adicionar a cada passo, a cidade k (ainda não visitada) entre a ligação (i, j) de cidades já visitadas, cujo custo de inserção seja o mais barato

Heurística da Inserção Mais Barata

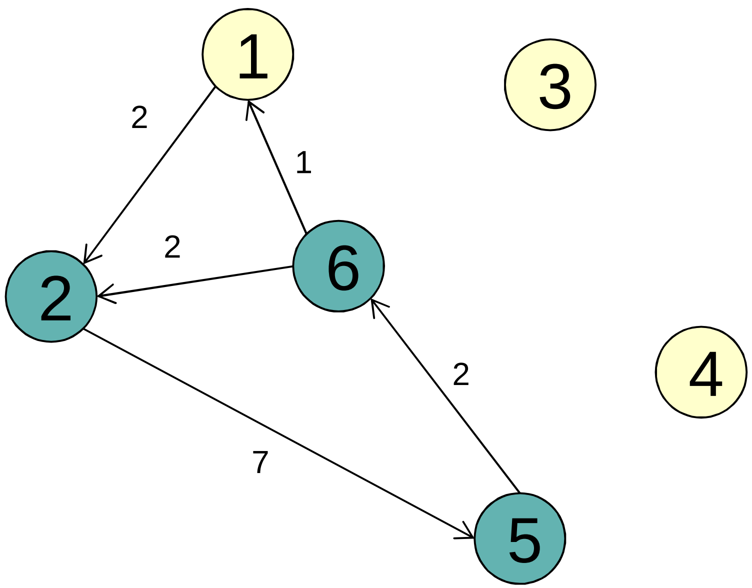
$$\textbf{Custo da inserção} = d_{ik} + d_{kj} - d_{ij}$$



PCV – Inserção mais Barata :: Exemplo - Passo 2

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	2
4	4	9	3	0	2	3
5	9	7	8	2	0	2
6	1	2	2	3	2	0

i	k	j	$d_{ik} + d_{kj} - d_{ij}$
6	**1**	**2**	1 + 2 – 2 = 1
6	**3**	**2**	2 + 5 – 2 = 5
6	**4**	**2**	3 + 9 – 2 = 10
2	**1**	**5**	2 + 9 – 7 = 4
2	**3**	**5**	5 + 8 – 7 = 6
2	**4**	**5**	9 + 2 – 7 = 4
5	**1**	**6**	9 + 1 – 2 = 8
5	**3**	**6**	8 + 6 – 2 = 12
5	**4**	**6**	2 + 3 – 2 = 3

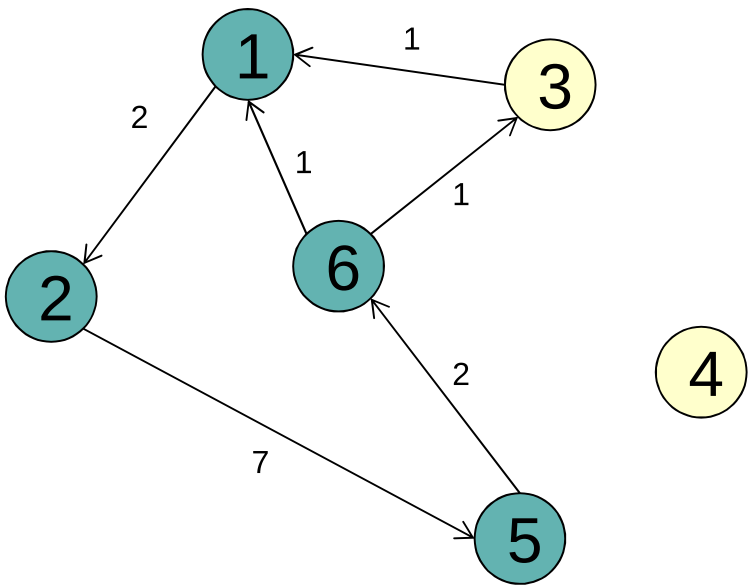


11 + 1 = 12

PCV – Inserção mais Barata :: Exemplo - Passo 3

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	2
4	4	9	3	0	2	3
5	9	7	8	2	0	2
6	1	2	2	3	2	0

i	k	j	$d_{ik} + d_{kj} - d_{ij}$
6	**3**	**1**	2 + 1 – 1 = 2
6	**4**	**1**	3 + 4 – 1 = 8
1	**3**	**2**	1 + 5 – 2 = 4
1	**4**	**2**	4 + 9 – 2 = 11
2	**3**	**5**	5 + 8 – 7 = 6
2	**4**	**5**	9 + 2 – 7 = 4
5	**3**	**6**	8 + 2 – 2 = 8
5	**4**	**6**	2 + 3 – 2 = 3

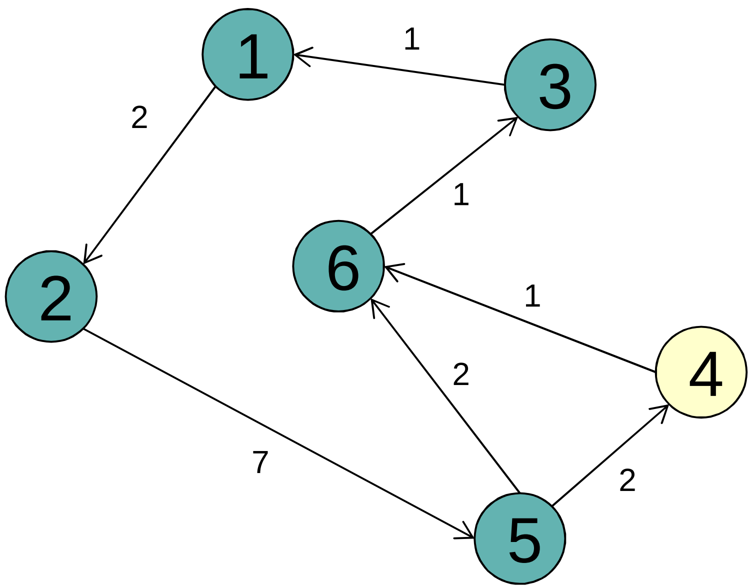


12 + 2 = 14

PCV – Inserção mais Barata :: Exemplo – Passo final

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	2
4	4	9	3	0	2	3
5	9	7	8	2	0	2
6	1	2	2	3	2	0

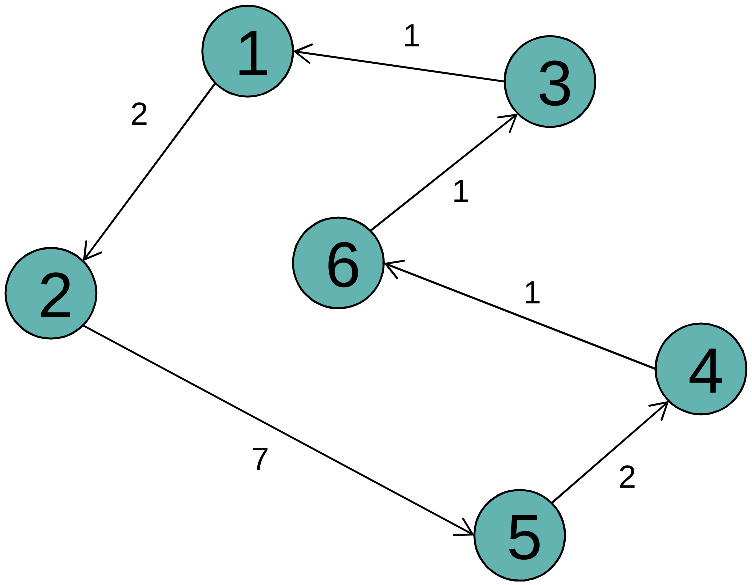
i	k	j	$d_{ik} + d_{kj} - d_{ij}$
6	**4**	**3**	3 + 3 – 2 = 4
3	**4**	**1**	3 + 4 – 1 = 6
1	**4**	**2**	4 + 9 – 2 = 11
2	**4**	**5**	9 + 2 – 7 = 4
5	**4**	**6**	2 + 3 – 2 = 3



14 + 3 = 17

PCV – Inserção mais Barata :: Exemplo – Passo final

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	2
4	4	9	3	0	2	3
5	9	7	8	2	0	2
6	1	2	2	3	2	0

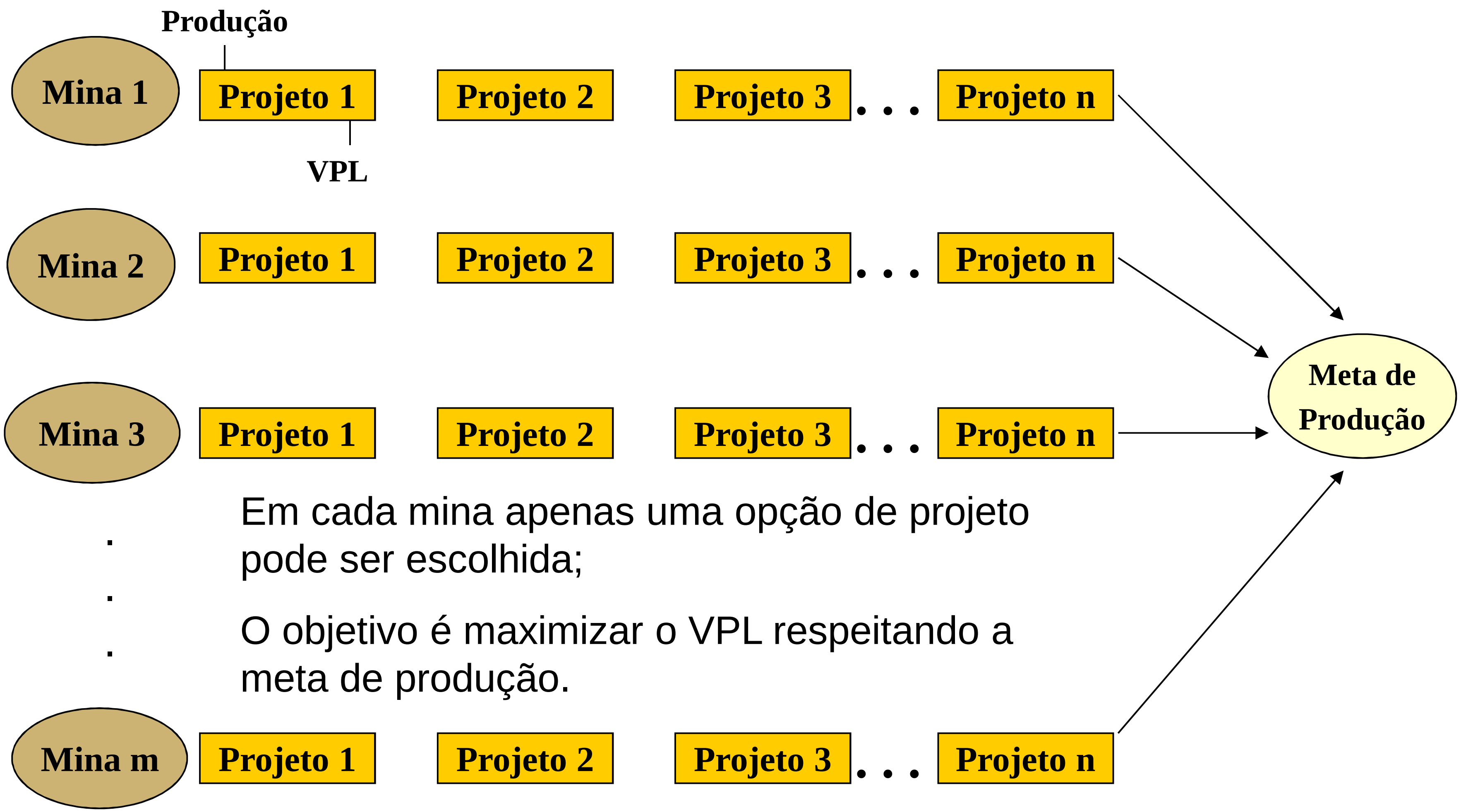


- **Distância Total *s* =** 17

S = (6 3 1 2 5 4)

PROBLEMA DA SELEÇÃO DE PROJETOS

Seleção de Projetos Concorrentes



Em cada mina apenas uma opção de projeto pode ser escolhida;

O objetivo é maximizar o VPL respeitando a meta de produção.

Seleção de Projetos Concorrentes

Opção	Produção	VPL ($\$x10^6$)	Custo/t.	Observações:
1	40.00	884.42	4.59	Caso Base.
2	43.00	929.56	4.36	Aumento Capacidade.
3	43.00	922.04	5.08	Aumento Capacidade c/risco operacional.
4	43.00	885.11	5.16	Aumento Capacidade c/ aumento custo.
5	40.00	857.75	4.39	Caso Base com variação de risco.
6	40.00	874.16	4.61	Caso Base com variação de risco.
7	40.00	1000.00	4.42	Caso Base com variação de risco.
8	40.00	857.07	4.56	Caso Base com variação de risco.
9	40.00	804.41	4.63	Caso Base com variação de risco.
10	42.00	967.17	4.70	Aumento Capacidade 2.
11	42.00	909.04	4.75	Aumento Capacidade 2 c/ risco.
12	42.00	843.39	4.84	Aumento Capacidade 2 c/ risco.
13	48.00	972.64	3.88	Aproveitamento Minério Marginal.
14	48.00	935.71	3.90	Ídem c/ variação de investimento.
15	45.00	930.93	4.14	Ídem c/ variação de produção.
16	45.00	897.42	4.16	Ídem c/ variação de investimento.

Seleção de Projetos Concorrentes

Exemplo de VPL ($\$ \times 10^6$):

	1	2	3	...	16
1	884.42	929.56	922.04	...	897.42
2	177.28	149.92	199.85	...	247.04
...	...	...	...	...	...
7	111.63	106.84	103.42	...	126.67

Exemplo de Produção (Mt):

	1	2	3	...	16
1	40	43	43	...	45
2	10	10	10	...	12.5
...	...	...	...	...	...
7	2	2	2	...	4

Seleção de Projetos Concorrentes

Dados de entrada:

- Minas: conjunto de minas;
- Projetos: conjunto de opções de projeto em cada mina;
- VPL_{ij}: Valor Presente Líquido do projeto j para a mina i;
- $prod_{ij}$: produção do projeto j na mina i;
- M: meta de produção;
- Pfalta: penalidade por produção inferior à meta estabelecida;
- Pexc: penalidade por produção superior à meta estabelecida.

Seleção de Projetos Concorrentes

Variável de Decisão:

- $$x_{ij} = \begin{cases} 1 \text{ caso a mina i opere com o projeto j} \\ 0 \text{ caso contrário} \end{cases}$$

Seleção de Projetos Concorrentes

$$\max \sum_{i \in Minas} \sum_{j \in Projetos} VPL_{ij} x_{ij} - Pfalta \times dnp - Pexc \times dpp$$

$$\sum_{j \in Projetos} x_{ij} = 1 \ \forall i \in Minas$$

$$\sum_{i \in Minas} \sum_{j \in Projetos} prod_{ij} x_{ij} - dpp + dnp = M$$

$$x_{ij} \in \{0,1\} \ \forall i \in Minas, \ \forall j \in Projetos$$

HEURÍSTICA

PROBLEMA DA SELEÇÃO DE PROJETOS

Seleção de Projetos Concorrentes

Representação de uma solução:

7	15	7	3	8	14	5
1	2	3	4	5	6	7

*Obs.: Com esta representação, a restrição de que em cada mina um projeto tem que ser implementado é **automaticamente** satisfeita

Heurísticas de refinamento

<u>Princípio de funcionamento:</u>

- Partir de uma solução inicial qualquer
- Caminhar, a cada iteração, de vizinho para vizinho de acordo com a definição de vizinhança adotada, tentando **<u>melhorar</u>** a solução construída

Seleção de Projetos Concorrentes

Exemplo de vizinhança:

Solução *s*:

7	15	7	3	8	14	5
1	2	3	4	5	6	7

Exemplo de vizinho *s*': gerado a partir de s, substituindo-se o projeto implementado em uma mina por outro

7	15	7	3	10	14	5
1	2	3	4	5	6	7

Método da descida/subida (Descent/Uphill Method)

- Parte de uma solução inicial qualquer
- A cada passo analisa **todos** os possíveis vizinhos
- Move somente para o vizinho que representa uma **melhora** no valor atual da função de avaliação.
  - Em problemas de minimização, um vizinho *s'* é melhor que *s* quando f(*s'*) < f(*s*)). Neste caso, *s'* passa a ser a nova solução corrente
- O método é interrompido quando um ótimo local (com respeito à definição de vizinhança) é encontrado

METAHEURÍSTICAS

Metaheurísticas

- Métodos heurísticos, de caráter geral, dotados de mecanismos para escapar das armadilhas dos ótimos locais
- Princípio básico: aceitar soluções de piora
- Podem ser baseados em Busca Local ou Busca Populacional.
- Os métodos baseados em Busca Local são fundamentados na noção de vizinhança:
  - Dada uma solução s, diz-se que s' é um vizinho de s, se s' é obtido de s a partir de um movimento m, isto é: $s' \leftarrow s \oplus m$
  - A estrutura de vizinhança varia de acordo com o problema tratado
- Os métodos baseados em Busca Populacional partem de um conjunto de soluções, aplicando sobre estes operadores que visam à melhoria desse conjunto.

Metaheurísticas

- Exemplos de metaheurísticas:
  - de busca local:
    - Busca Tabu
    - *Simulated Annealing*
    - *Iterated Local Search* (ILS)
    - *Variable Neighborhood Search* (VNS)
  - de busca populacional:
    - Algoritmos Genéticos
    - Colônia de Formigas

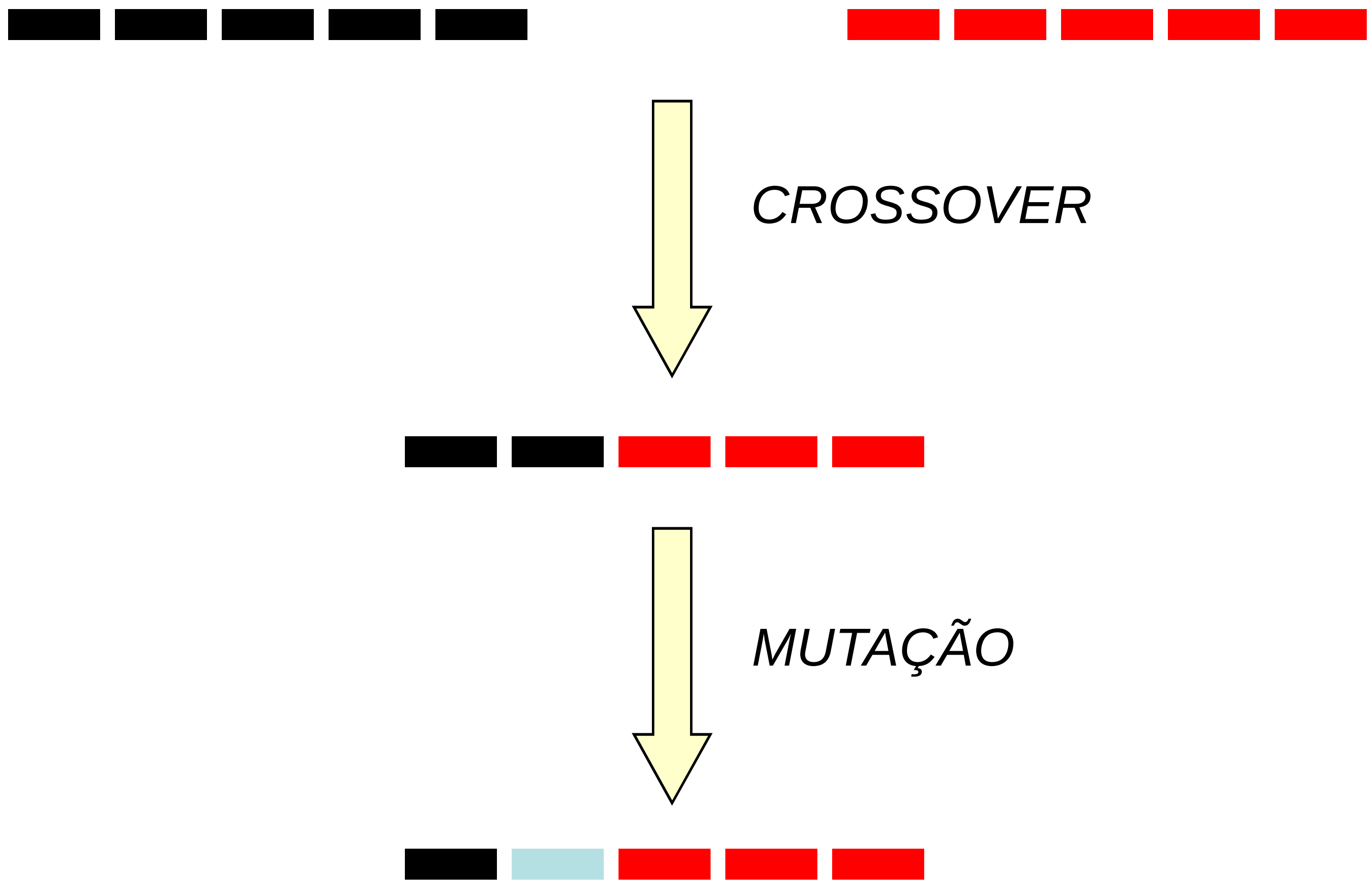
Algoritmo Genético

- Os Ags fazem uma analogia com processos naturais de evolução:
- Dada uma população, os indivíduos com características genéticas melhores têm maiores chances de sobrevivência e de produzirem filhos cada vez mais aptos, enquanto indivíduos menos aptos tendem a desaparecer
- As características dos indivíduos, registradas em seus genes, são transmitidas para seus descendentes e tendem a propagar-se por novas gerações

Algoritmo Genético

- Características dos descendentes são parcialmente herdadas de seus pais (**Crossover**) e parcialmente de novos genes criados durante o processo de reprodução (**Mutação**)
- O objetivo de um AG é tentar melhorar as qualidades genéticas de uma população através de um processo de renovação iterativa das populações

Operadores genéticos



Relação entre AG e Problema de Otimização

AG	Problema de Otimização
Indivíduo	Solução de um problema
População	Conjunto de soluções
Cromossomo	Representação de uma solução
Gene	Parte da representação de uma solução
Alelos	Valores que uma variável pode assumir
Crossover / Mutação	Operadores de busca

Avaliação de cromossomos

- Feita pela função de aptidão (**fitness**)
- Em um problema de maximização pode ser a própria função objetivo
- Em um problema de minimização pode ser o inverso da função objetivo

Fase de seleção

- Binary tournament selection:
  - Selecionar dois indivíduos aleatoriamente
  - O primeiro pai é o indivíduo com maior aptidão
  - Selecionar, aleatoriamente, outros dois pais
  - O segundo pai é o indivíduo com maior aptidão nessa nova seleção
- Aleatório
- Roleta russa

Fase de reprodução

- Operador **mutação** clássico
- Dois ou mais cromossomos passam por um processo de mutação e/ou recombinação para gerar novos cromossomos filhos (*offsprings*)

7	*15*	*7*	*3*	*10*	*14*	*5*
1	2	3	4	5	6	7

7	*15*	*7*	*3*	*10*	*14*	*3*
1	2	3	4	5	6	7

Fase de reprodução

- Operador **crossover** clássico (one point crossover):
- Descendentes são formados a partir da reunião de segmentos de cada pai:

7	*15*	*7*	*3*	*10*	*14*	*5*
1	2	3	4	5	6	7

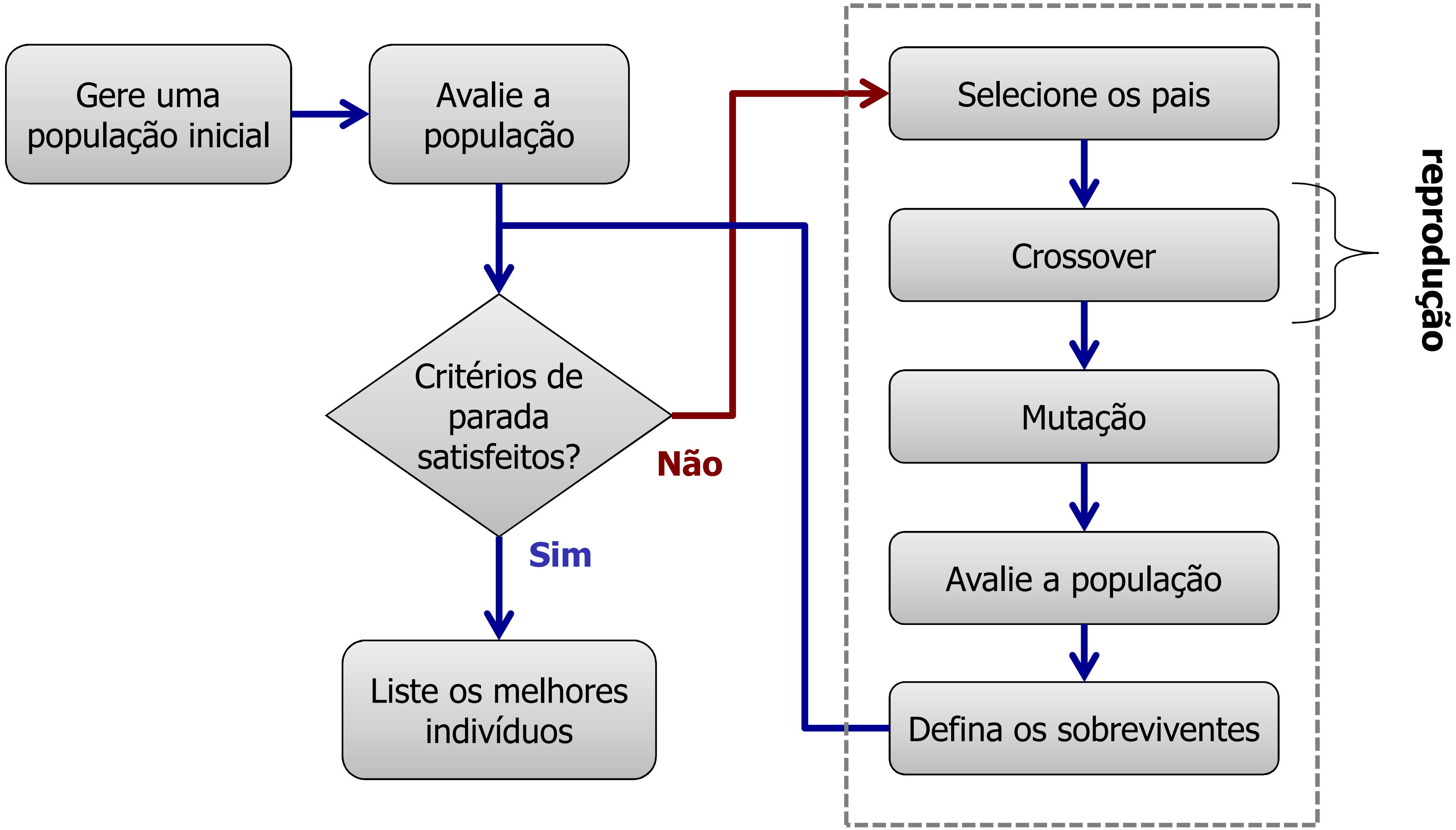
12	*3*	*8*	*5*	*10*	*4*	*2*
1	2	3	4	5	6	7

7	*15*	*7*	*5*	*10*	*4*	*2*
1	2	3	4	5	6	7

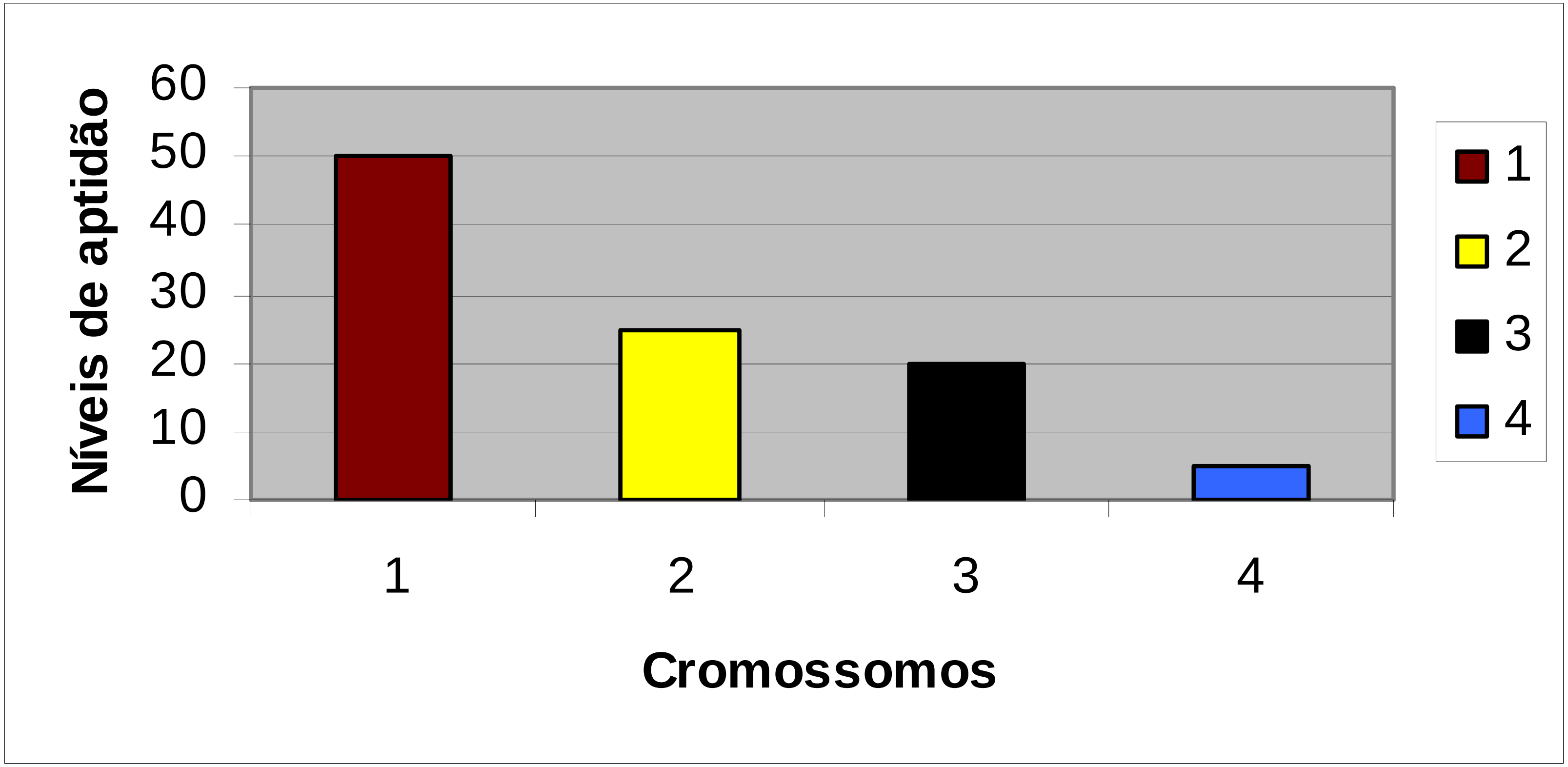
Sobrevivência / morte de cromossomos

- Como selecionamos os cromossomos que devem sobreviver?
- Sobrevivem os que possuem os melhores níveis de aptidão?
- É importante permitir também a sobrevida de cromossomos menos aptos, do contrário o método ficaria preso em ótimos locais
- Elitismo

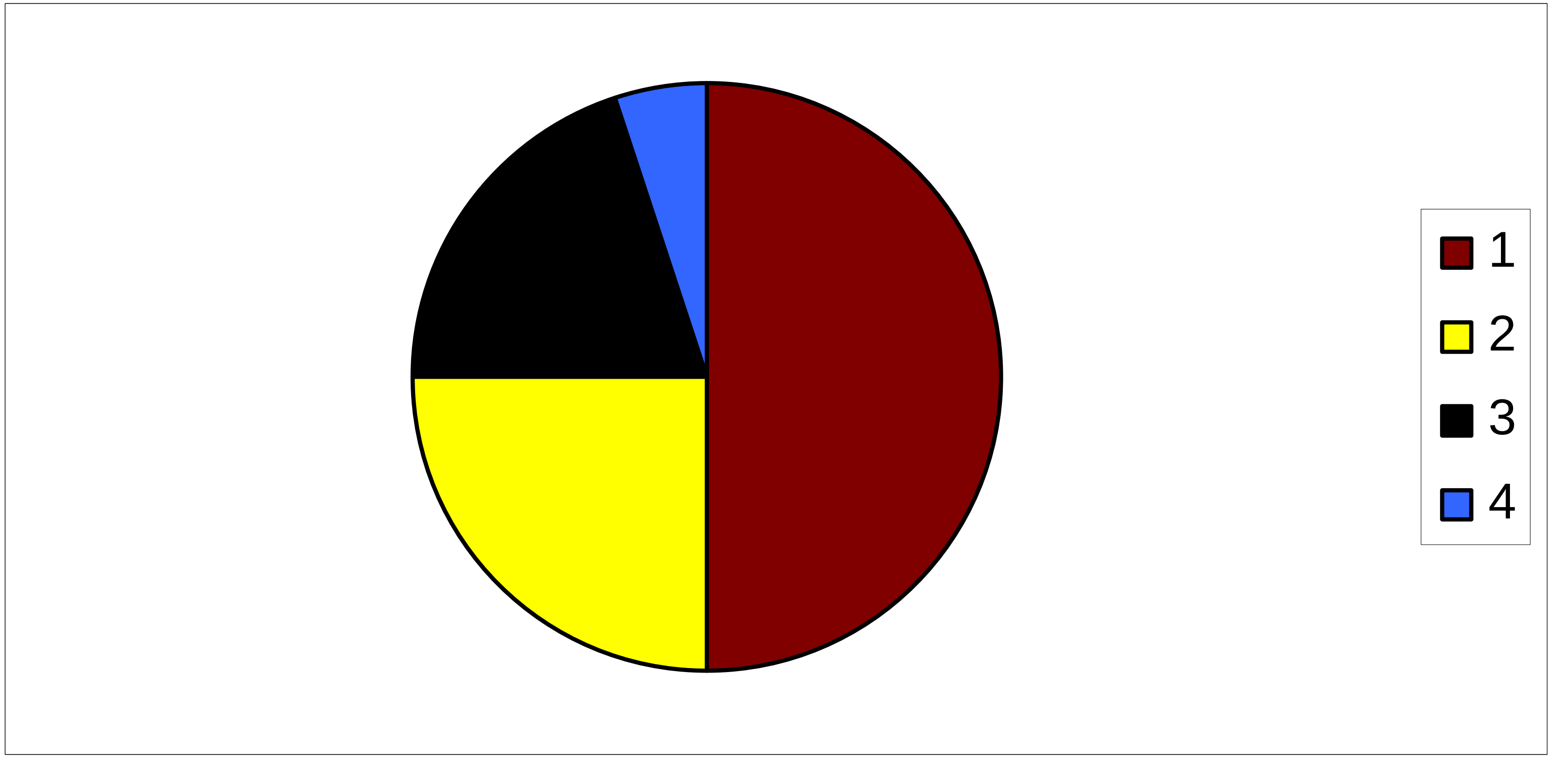
Algoritmo Genético



Seleção de cromossomos sobreviventes



Roleta Russa (Seleção de sobreviventes)



Iterated Local Search (ILS)

- Método de busca local
- Explora o espaço de soluções por meio de perturbações nas soluções ótimas locais, seguidas de procedimentos de busca local
- Se ao aplicar a busca local não for encontrada uma solução melhor, a perturbação é aumentada
- Sempre que é encontrada uma solução melhorada, a perturbação volta ao nível mais mínimo
- Termina quando um critério de parada for atingido, p. ex., tempo de processamento

Iterated Local Search (ILS)