Modelagem e Otimização Algorítmica

Visão Geral da Disciplina

Prof. Dr. Ademir Aparecido Constantino

Departamento de Informática

Universidade Estadual de Maringá

http://www.din.uem.br/~ademir

ademir@din.uem.br

Moodlep

Senha: MOA2020

Não está autorizada a gravação das aulas desta disciplina.

Programa

- Introdução
  - Apresentar casos reais usando programação matemática e algoritmos heurísticos
  - Ilustrar casos de automação operacional por software
  - Programação matemática e modelos baseados em grafos
  - Ilustrar casos

Programa

- Programação Matemática
  - Modelagem de problemas (formulação)
  - Resolução computacional de problemas de programação linear
  - Algoritmo para resolução de programação linear (introdução ao método simplex)
  - Formulação e resolução de problemas de programação linear inteira
  - Introdução ao método *branch-and-bound*
    - Problema de transporte
    - Problema de designação
    - Problema do caixeiro viajante
    - Problema da mochila
    - Problema de cobertura de conjuntos
    - Outros problemas
  - Apresentar ferramentas comerciais e software livre para programação matemática

Programa

Programação Dinâmica (projeto de algoritmos)

- Programação Dinâmica
- Elementos da programação multi-estágios
- Modelos recursivos
- O paradigma para projeto de algoritmo
- Complexidade computacional do paradigma de programação dinâmica
- Aplicações

Programa

Algoritmos Heurísticos

- Definição de algoritmo heurístico
- Representação computacional de soluções
- Vizinhanças, espaço de busca, ótimo global e ótimo local
- Classificação heurísticas
- Algoritmos heurísticos construtivos (algoritmo guloso)
- Algoritmos de busca local (*hill climbing*)
- Algoritmo A*
- Aplicações

Programa

- Meta-heurística (projeto de algoritmos heurísticos)
  - Introdução, definições e taxinomia
  - Busca na Vizinhança Variável (VNS)
    - Estrutura de vizinhança
    - Estrutura de um algoritmo VNS e VND
    - Aplicações
  - GRASP: *Greedy Randomized Adaptive Search Procedures*
    - Algoritmos *semi-greedy*
    - Escolha de candidatos: *restricted candidate list* (RCL)
    - Busca local a partir da solução *semi-greedy*
    - Calibragem da RCL (método reativo e método bias)
    - Aplicações
  - *Simulated Annealing*
    - Componentes principais
    - Parâmetros do algoritmo: temperatura (inicial e final), taxa de resfriamento, etc
    - Esquemas de arrefecimento e sua implicação no desempenho do algoritmo
    - Aplicações

Programa

- Sistemas Fórmicos (ACO – *Ant Colony Optimization)*)
  - Analogia com o comportamento das formigas
  - Analogia com o paradigma guloso
  - Elementos de algoritmos baseados em ACO
  - Aplicações
- Algoritmos Genéticos (AG) e Algoritmos Meméticos
  - Elementos de algoritmos baseados em AG
  - Representação das soluções (indivíduos)
  - Operadores genéticos
  - Aplicações
- Busca Tabu
  - Conceito de intensificação e diversificação
  - Elementos de algoritmos baseados em Busca Tabu
  - Aplicações

AVALIAÇÃO

Avaliação 1 – Prova escrita (Peso 1)

Avaliação 2 – Trabalho Prático (Peso 1)

Avaliação 3 – Trabalho Prático(Peso 1)

Contextualização

Algoritmos I – Fundamentos de Algoritmos

Algoritmos II – Estrutura de Dados

Algoritmos III – Projeto e Análise de Algoritmos

Algoritmos IV – Algoritmos em Grafos

Algoritmos V – Modelagem e Otimização Algorítmica

O que é Programação?

O que é Codificação?

Coding Vs Programming

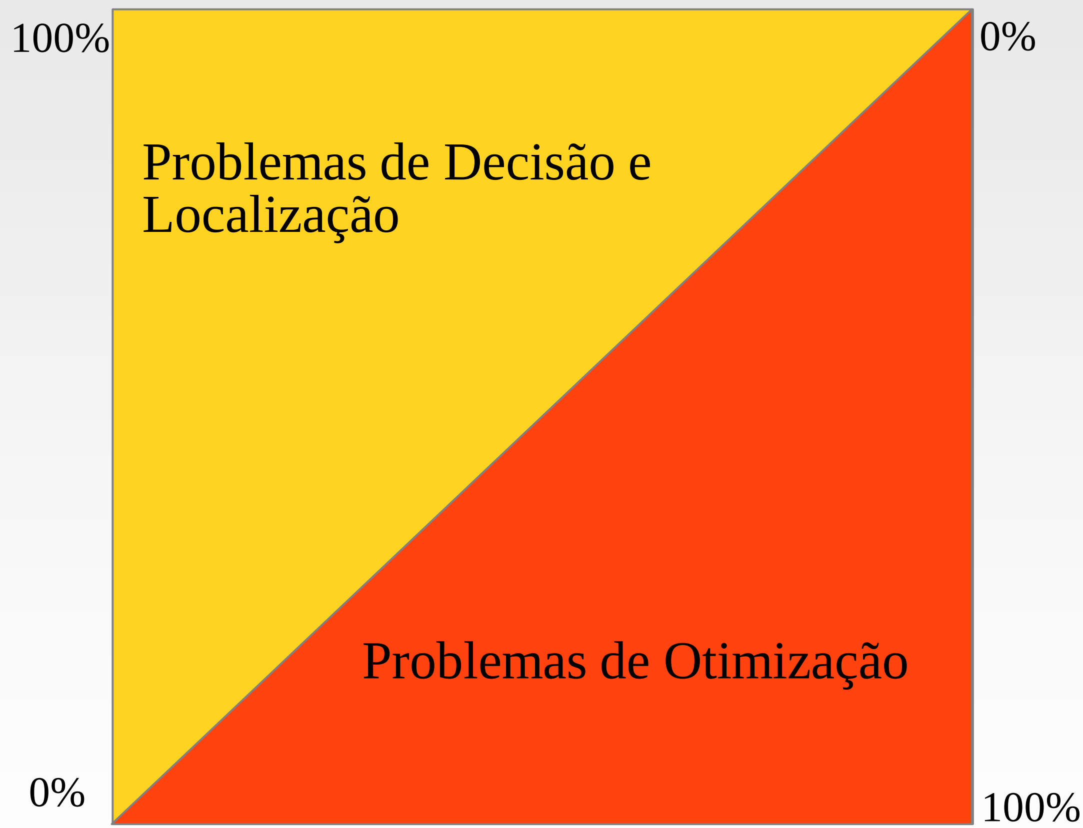
Problema X Algoritmo

- Quantos tipos de problemas podem ser resolvidos por algoritmos?
  - Resposta: a literatura classifica os problemas existentes em apenas 3 (três) tipos.
- *Tipos de problemas existentes:*
  - *Problema de Decisão;*
  - *Problemas de Localização;*
  - *Problemas de Otimização*

Distribuição do Conteúdo no Curso

Algoritmos I

Algoritmos V

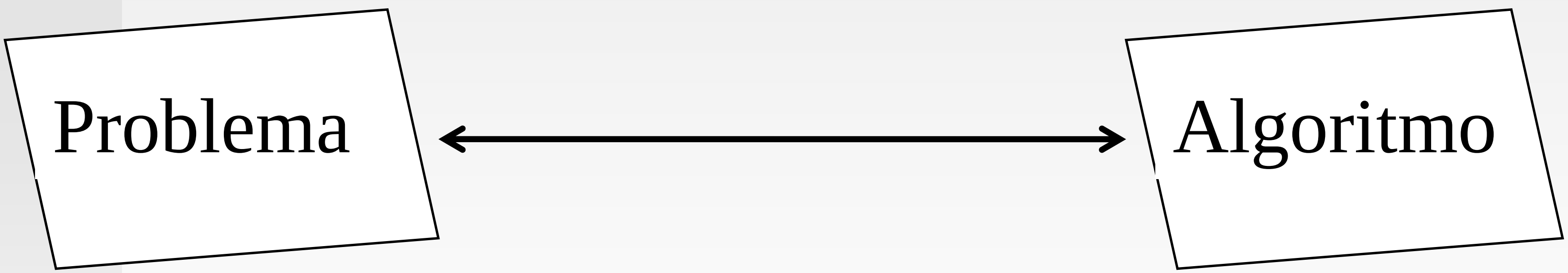


Tópicos

- Modelagem de Problemas
- Programação Matemática
  - Programação Linear
  - Programação Dinâmica
- Algoritmos Heurísticos
  - Algoritmo A*
  - Meta-heurísticas:
    - Algoritmos Genéticos
    - GRASP
    - Simulated Annealing
    - Ant System
    - Busca Tabu

Disciplinas Introdutória de Algoritmos

- O que você aprende?



Modelagem e Otimização Algorítmica

- O que você aprende?

Problema ⟷ Modelo ⟷ Algoritmo

Programação Matemática
Programação Dinâmica
Teoria dos Grafos
Teoria das Filas
etc.

Por que modelar problemas?

- Para muitos problemas de otimização a obtenção de algoritmos eficientes é extremamente difícil.
- Uma alternativa viável e importante é a modelagem desses problemas.
- A modelagem facilita a obtenção de algoritmos mais eficientes para os problemas de otimização.

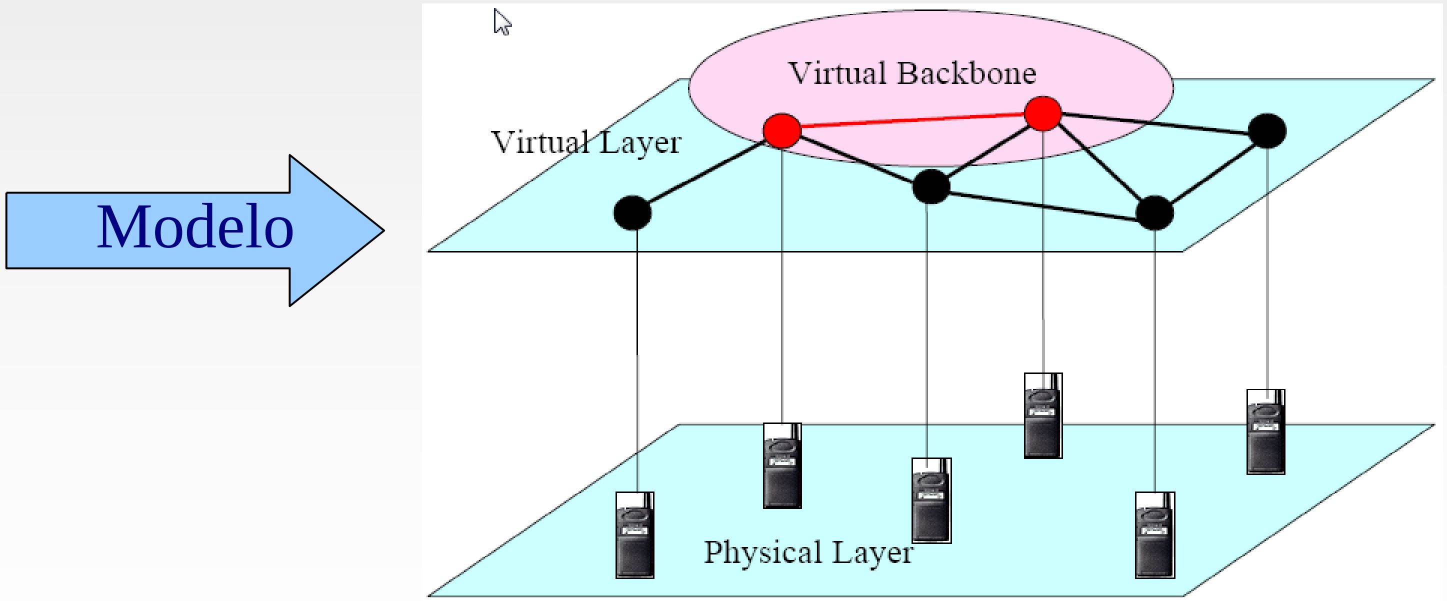
Como modelar um problema real

- Desenvolver (criar) um modelo baseado em alguma técnica conhecida, como programação matemática, grafos, etc.
  - –Técnica + arte + treinamento

- Utilizar algum modelo “teórico/análogo” já conhecido na literatura.
  - Exemplo:
    - ☞Problema de coloração para construção de horário escolar;
    - ☞Problema de cobertura de conjunto para escalonamento de trabalho.

Exemplos de modelagem por grafos

- Localização de *backbone* em rede *ad hoc* sem fio.

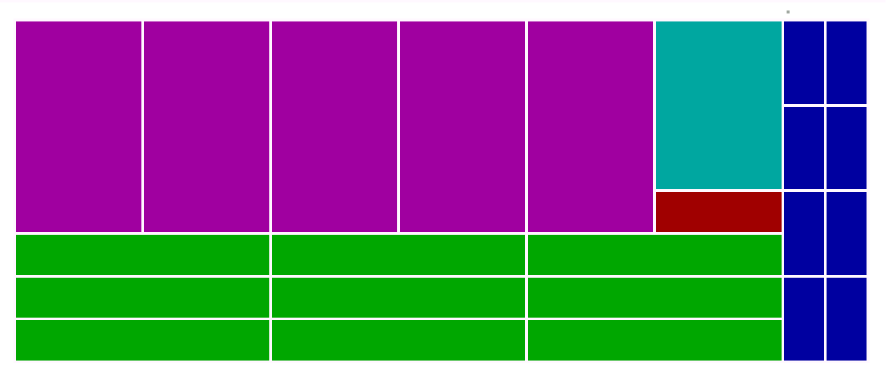


Localização de *backbone* de tamanho mínimo:

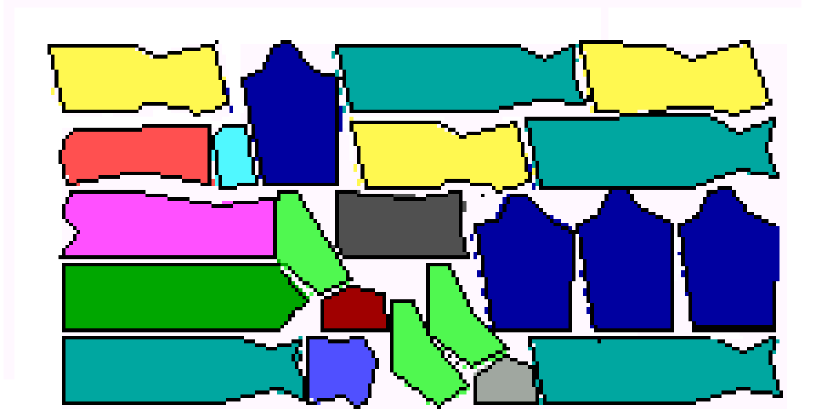
Conjunto Conexo Dominante Mínimo

Aplicações em Pesquisa Operacional

- Corte de Materiais
  - ☞Corte de peças regulares:

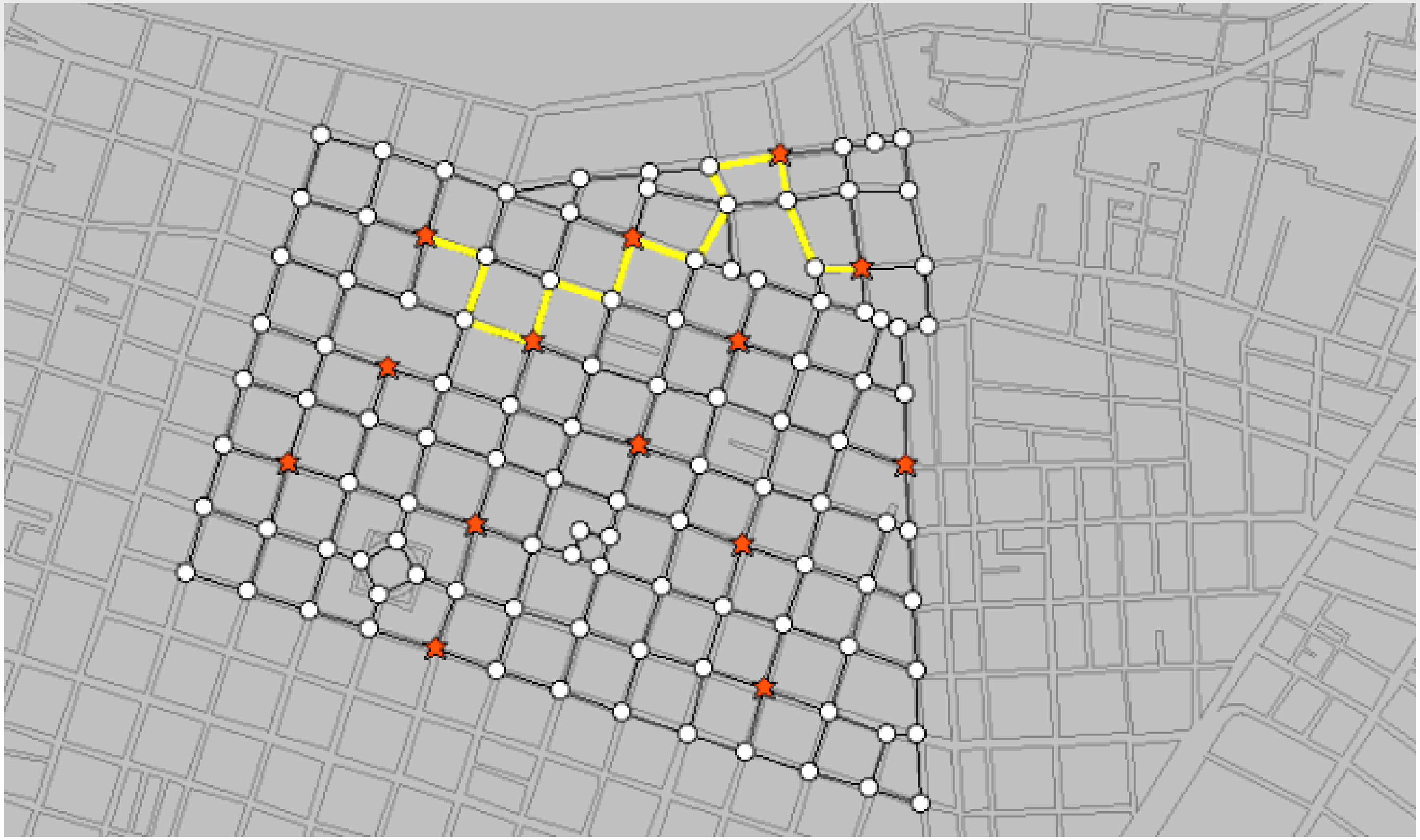


  - ☞Corte de peças irregulares:



Aplicações em Pesquisa Operacional (cont.)

- Distribuição física de produtos (logística)



Aplicações em Pesquisa Operacional (cont.)

- Escalonamento de mão-de-obra e tarefas
  - Dados: um conjunto de tarefas a ser realizado e um conjunto de funcionários
  - Objetivo: encontrar a melhor maneira de alocar os funcionários às tarefas de forma que todas as tarefas sejam cumpridas e os gastos com mão-de-obra sejam minimizados.
  - Restrições:
    - leis trabalhistas
    - restrições operacionais da empresa.

Conhecem um algoritmo exato para resolver o

Problema do Caixeiro Viajante?

Exemplo: Problema do Caixeiro Viajante

min $\left(\sum_{i}^{n}\sum_{j}^{n} d_{ij}\cdot X_{ij}\right)$ →minimizar o percurso total

X_{ij} = 1 se a aresta (i,j) for utilizada, 0 caso contrário.

suj. a:

(1) cada uma das cidades é visitada uma e só uma vez, ou seja, cada vértice é entrado uma só vez e saído uma só vez:

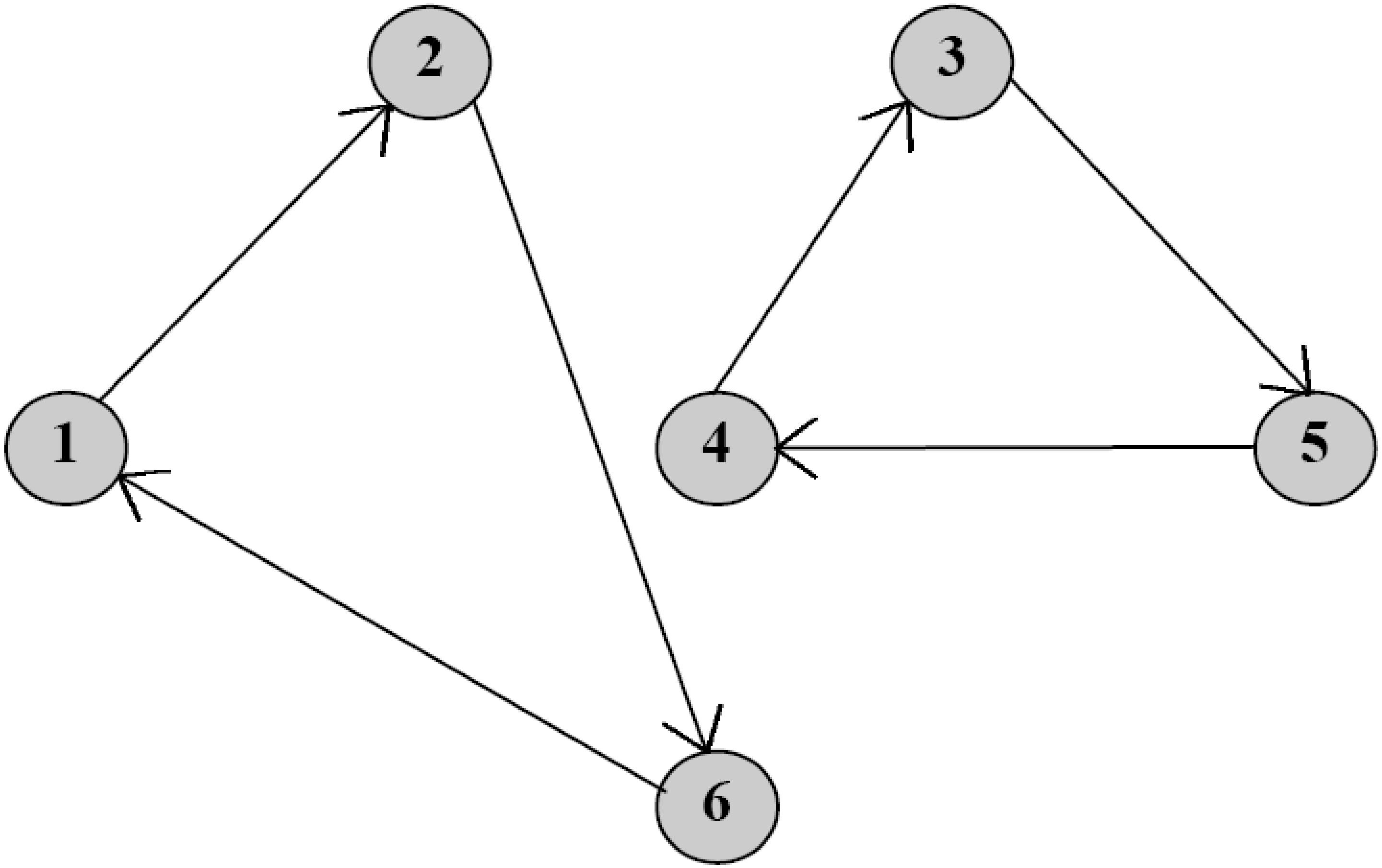
$$\sum_{j}^{n} X_{ij} = 1, \quad \forall_{\mathbf{i:\ i=1,..,n}}$$

$$\sum_{i}^{n} X_{ij} = 1, \quad \forall_{\mathbf{j:\ j=1,..,n}}$$

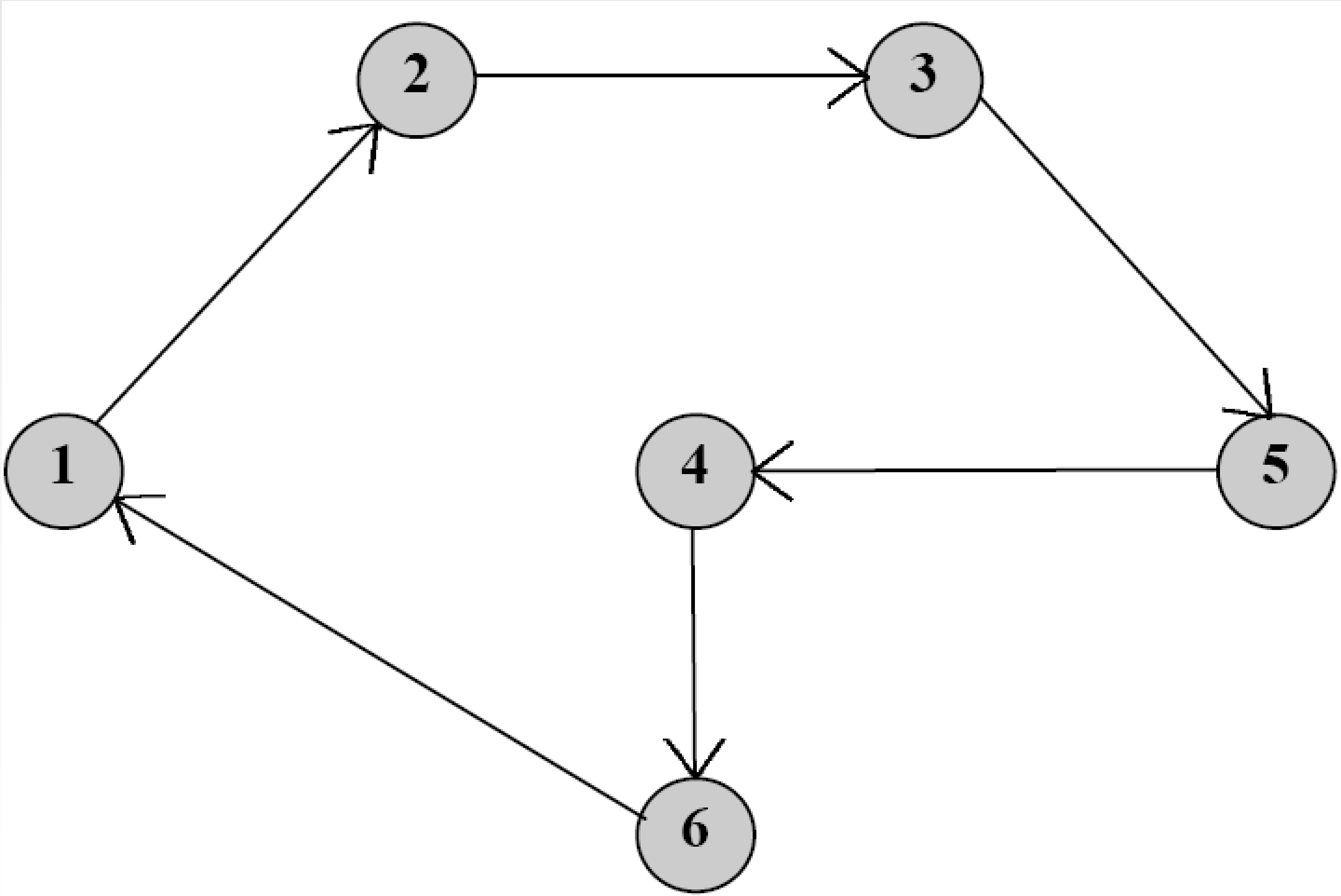
(2) entre dois quaisquer subconjuntos complementares de cidades (S e $\overline{\text{S}}$) há pelo menos um arco de ligação:

$$\sum_{i\in S}\sum_{j\in \overline{S}} X_{ij} \geq 1, \quad \forall_{S\subset \text{ conjunto total das cidades a visitar}}$$

Restrição 1 não garante a solução



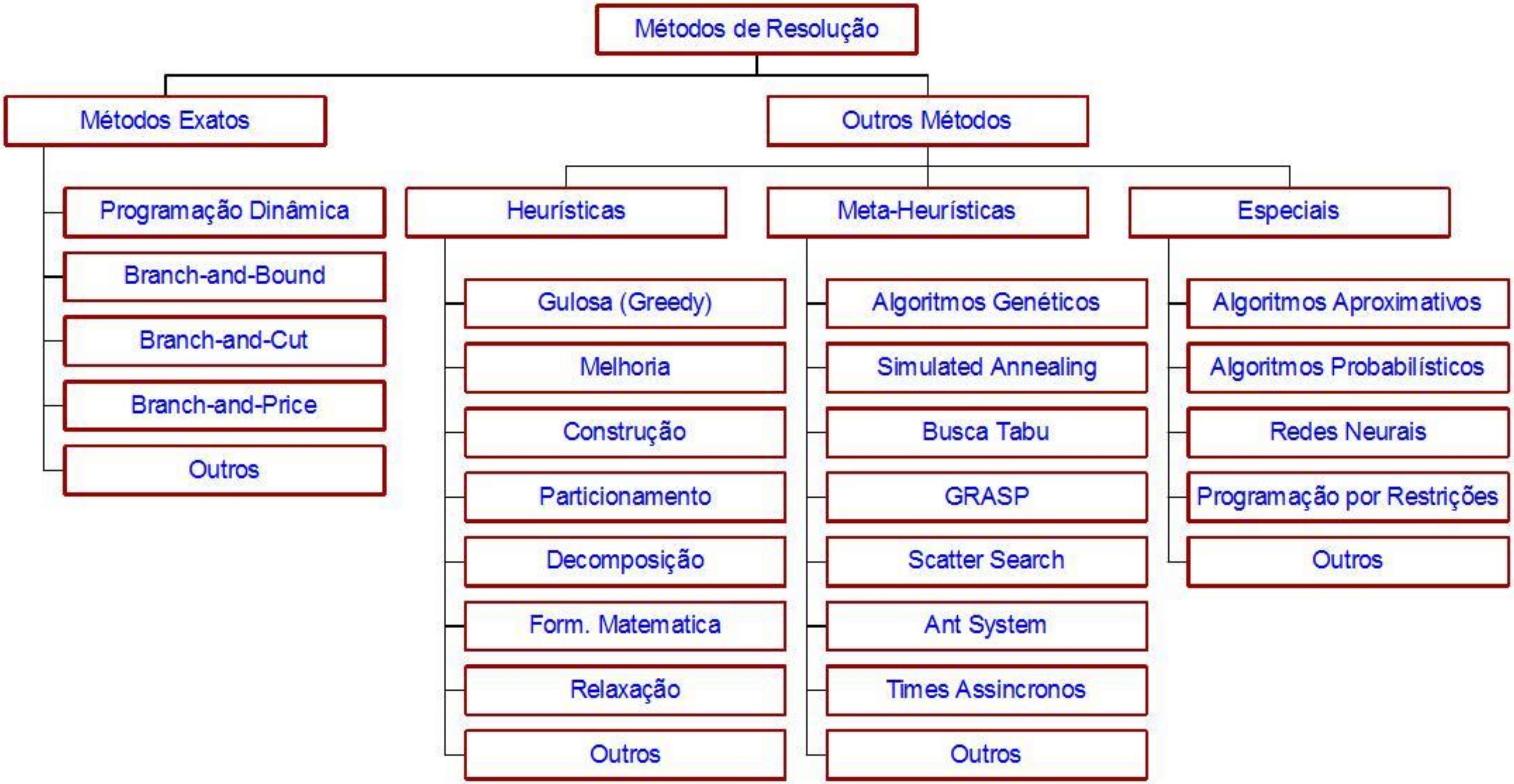
Restrição 2 não permite a formação de sub-circuitos disjuntos.



Desafios da área:

1. Modelagem de problemas;
2. Implementar algoritmos eficientes para resolver problemas em tempo computacional aceitável.

Como resolver os problemas de Otimização?



Exemplos de aplicações reais, usando técnicas que serão vistas durante este curso:

1 –Alocação automatizada de alunos em turmas em instituição de ensino superior.

2 –Geração automatizada de escalas de trabalho em empresas:

1. Escalonamento de Veículos em Empresas de Transporte Coletivo Urbano
2. Escalonamento de Motoristas em Transporte Coletivo Urbano

3 - Escalonamento de Maquinistas em uma Empresa Transporte Ferroviário de Carga.

1° Caso:

Alocação automatizada de alunos em turmas em instituição de ensino superior

Descrição do Problema

- Estudantes de graduação em regime seriado.
- Dois tipos de disciplinas são consideradas: a regular e a dependente.
- A disciplina regular é a disciplina programada na grade curricular vinculada à série que o aluno for se matricular.
- A disciplina em dependência é uma disciplina que o aluno não obteve aprovação em séries anteriores.

Descrição do Problema

- O principal objetivo: permitir que cada aluno assista o máximo de disciplinas possíveis dentre todas as disciplinas que lhe foram atribuídas previamente (regulares e dependência).
- Prioridades de Alocação:
  - baseada no desempenho e
  - baseada no grupo de pertinência.

Notação Matemática Utilizada

- K : número de cursos da instituição;
- n_k: número de alunos de um curso k, $k=1,...,K$;
- $D_a(i)$: conjunto de disciplinas que o aluno i deve cursar, $i=1,...,n_k$;
- $D_c(k)$ o conjunto de disciplinas de um curso k;
- $T_d(l)$: conjunto de turmas associadas a uma disciplina $l \in D_c(k)$;
- $T_a(i)=\{T_d(l): \forall\ l \in D_a(i)\}$: conjunto de todas as turmas que podem ser associadas ao aluno i;
- $T_c(k)=\{T_d(l): \forall\ l \in D_c(k)\}$: o conjunto de todas as turmas de um curso k;
- $t_\lambda(i)$: *conjunto de turmas conflitantes* do aluno i, $\lambda = 1,...,\eta_i$, onde η_i é o número de conjuntos de turmas conflitantes para o aluno i.

MDTC - Modelo para Distribuição de Turmas Baseado no Curso

$$Minimizar\left(p_1 \sum_{i=1}^{n_k} \sum_{j \in T_a(i)} c_{ij} x_{ij} + p_2 \sum_{l \in D_c(k)} g_l \right)$$

s.a. (a) $\sum_{j \in t_\lambda(i)} x_{ij} \leq 1; \quad \lambda = 1,\ldots,\eta_i; \; i = 1,\ldots,n_k;$

(b) $\sum_{j \in T_d(l)} x_{ij} = 1; \quad \forall l \in D_a(i); \; i = 1,\ldots,n_k;$

(c) $s_j = \sum_{i=1}^{n_k} x_{ij}; \quad \forall j \in T_c(k);$

(d) $s_j \leq g_l \quad \forall j \in T_d(l); \; \forall l \in D_c(k);$

(e) $x_{ij} = \begin{cases} 1 & \text{se a turma } j \text{ for selecionada para o aluno } i; \\ 0 & \text{caso contrário.} \end{cases}$

MDTA - Modelo para Distribuição de Turmas Baseado no Aluno

$$\textit{Minimizar} \sum_{j \in T_a(i)} c_j x_j$$

s.a. (a) $\displaystyle\sum_{j \in t_\lambda(i)} x_j \leq 1; \quad \lambda = 1, \ldots, \eta_i;$

(b) $\displaystyle\sum_{j \in T_d(l)} x_j = 1; \quad \forall l \in D_a(i)$

(c) $x_j = \begin{cases} 1 & \text{se a turma } j \text{ for selecionada} \\ 0 & \text{caso contrário.} \end{cases}$

$$c_j = \begin{cases} 5000 p_{ij} + 10 s_j & \text{se a turma } j \text{ corresponde a uma } \textbf{disciplina regular}; \\ 250.000\, p_{ij} + 10 s_j & \text{se a turma } j \text{ corresponde a uma } \textbf{disciplina em dependência}; \\ 10.000.000 & \text{se a turma } j \text{ for uma } \textbf{turma fictícia}. \end{cases}$$

Implementação

- Linguagem: C
- IBM R50 com um processador PowerPC 200MHz
- S.O.: Linux
- Resolvedor: **lp-solve** - um programa de código aberto baseado na licença Lesser GPL (*Lesser General Public License*).

Dados utilizados no teste

Descrição	2003	2004
Nº de cursos	31	31
Nº total de alunos	7.898	8.160
Nº médio de alunos por curso	254,77	263,23
Nº médio de disciplinas regulares por aluno	6,34	6,34
Nº médio de disciplinas em dependência por aluno	0,36	0,39

Resultados

S_UEM: o sistema da instituição;

S_MDTA: novo sistema proposto baseado no modelo MDTA.

Tempo de processamento (em segundos)

	2003		2004	
Tempo	S_UEM	S_MDTA	S_UEM	S_MDTA
Máximo	9.655	226	11.293	290
Mínimo	53	64	62	54
Médio	1202,06	113,03	1258,50	122,91
Total	37.262	3.504	39.014	3.810

Resultados

Número de alunos com conflito de horário em dependência

	2003		**2004**	
Nº alunos	S_UEM	S_MDTA	S_UEM	S_MDTA
Máximo	167	160	193	185
Mínimo	1	1	2	0
Médio	48,42	40,35	58,87	46,32
Total	1.501	1.251	1.825	1.436

Resultados

Número de alunos não alocados a turmas

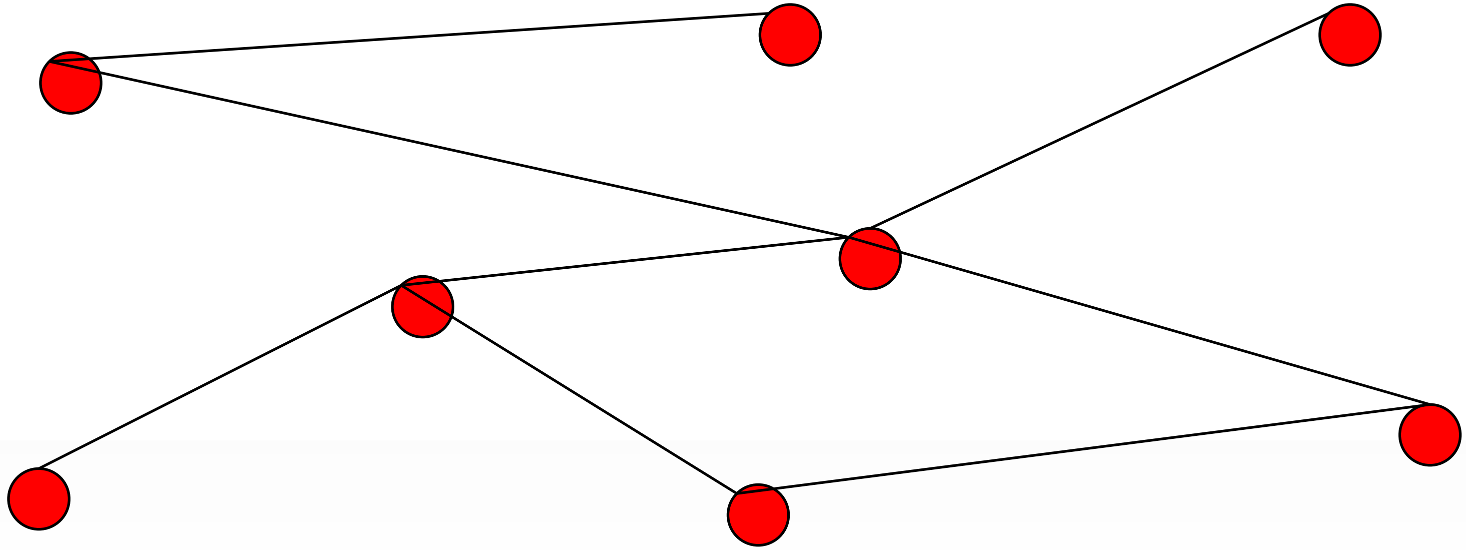
Nº Alunos	2003		2004	
	S_UEM	S_MDTA	S_UEM	S_MDTA
Máximo	58	23	58	6
Mínimo	0	0	0	0
Médio	14,10	3,42	8,19	0,58
Total	437	106	254	18

Conclusões

- Redução no tempo de processamento em aproximadamente 90%;
- Redução de trabalho manual (centenas de casos);
- Novas oportunidades de aulas aos alunos em dependência;
- Novo sistema passou a ser executado em horário de expediente;
- Mais agilidades nas tomadas de decisões.

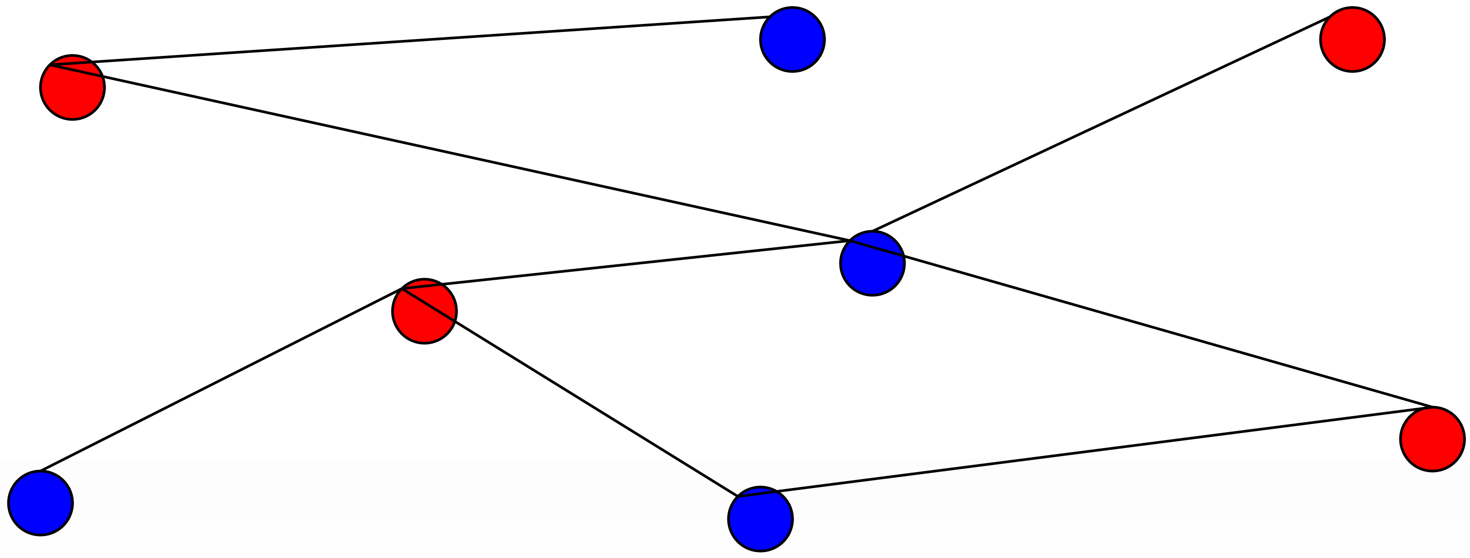
Nova abordagem sendo estudada

- Modelagem baseada em grafos
  - Aresta: Conflito de horário ou turmas da mesma disciplina;
  - Vértice: uma turma que o aluno pode ser alocado parte.



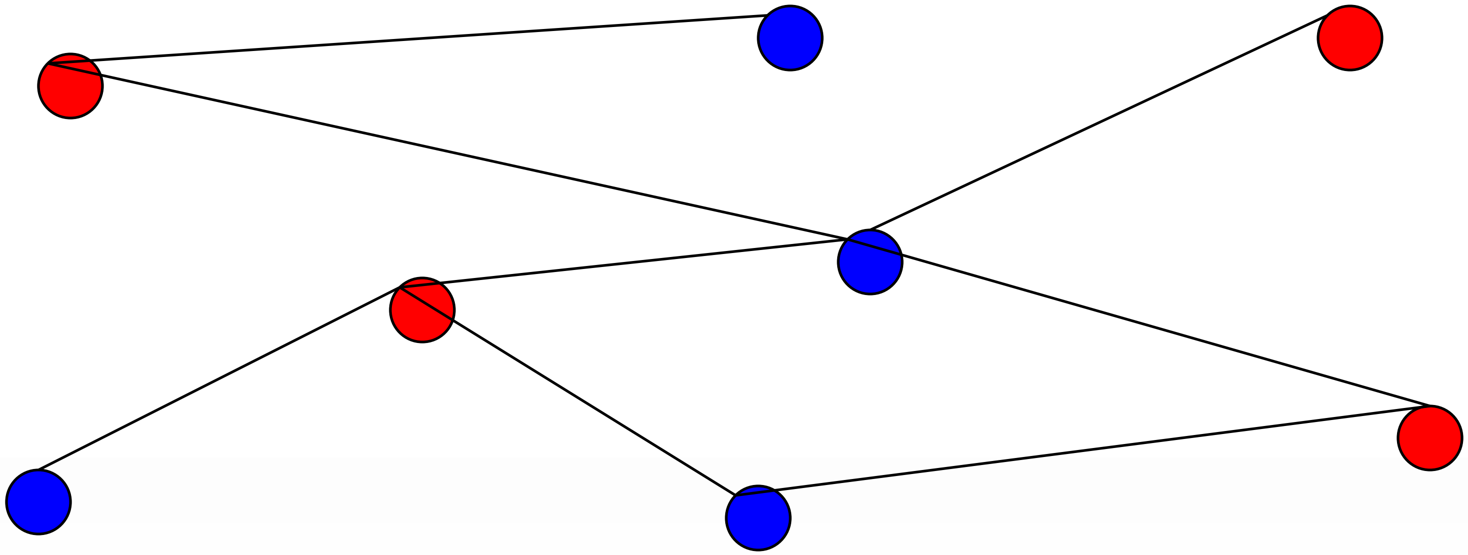
Modelagem baseada em grafos

- Solução:
  - Conjunto estável ou conjunto independente.



Modelagem baseada em grafos

- Solução:
  - Conjunto estável ou conjunto independente: é um subconjunto de vértice de tal maneira que não exista aresta entre eles.



2° Caso:

Geração automatizada de escalas de trabalho em empresas:

2.1. Escalonamento de Veículos em Empresas de Transporte Coletivo Urbano

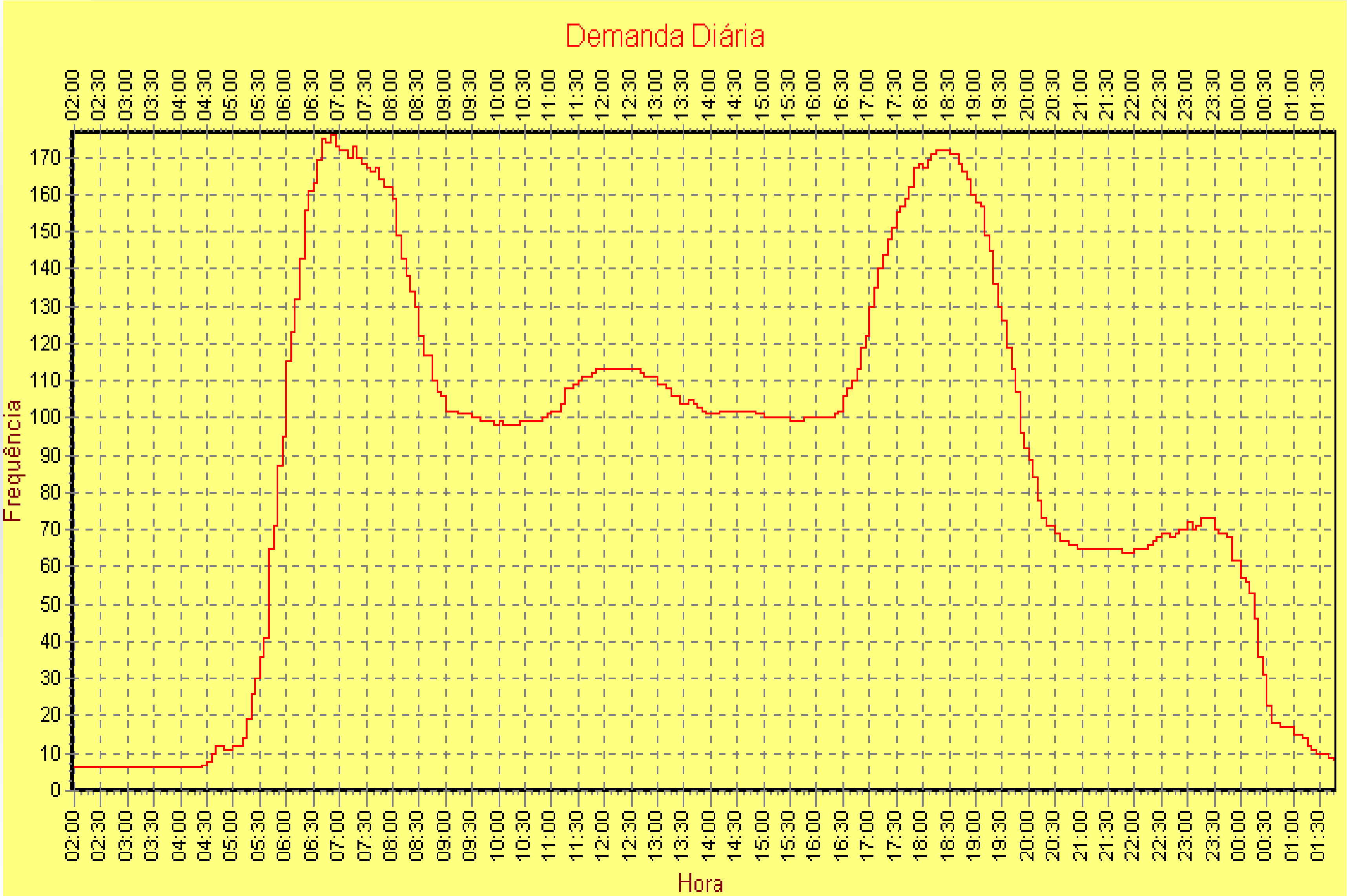
Problema

- Dados:
  - a tabela de horários com as viagens que a empresa deve cumprir;
  - a rede viária com os pontos de troca de condutor, as garagens e os estacionamentos;

Problema

- Dados:
  - a escala de viagens para os veículos;
  - as restrições trabalhistas;
    - Exemplos:
      - trabalho contínuo $\leq$ 5 horas
      - descanso mínimo = 1 hora
      - turno normal $\leq$ 8 horas
- Objetivo:
  - gerar seqüências de trabalho para os condutores de tal maneira que todas as viagens sejam cobertas, utilizando o número mínimo de condutores.

Distribuição das Viagens Durante um Dia Útil



Resolução do Problema

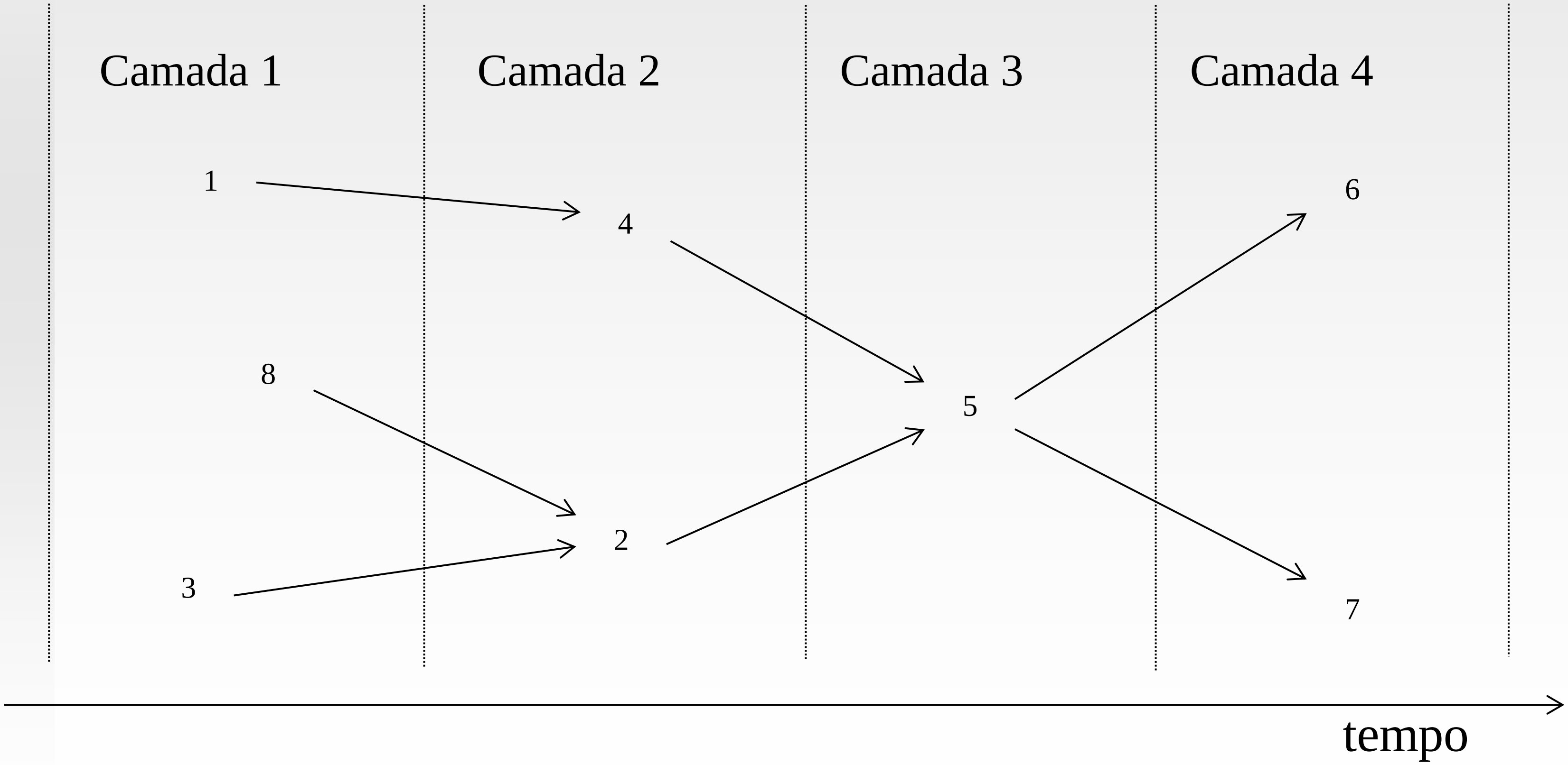
- Tendo em vista a complexidade computacional do problema, adotou-se um modelo heurístico.
- Estratégia utilizada:
  - Dividir o conjunto de viagens em camadas;
  - Resolver o emparelhamento entre as camadas.

Algoritmo Proposto

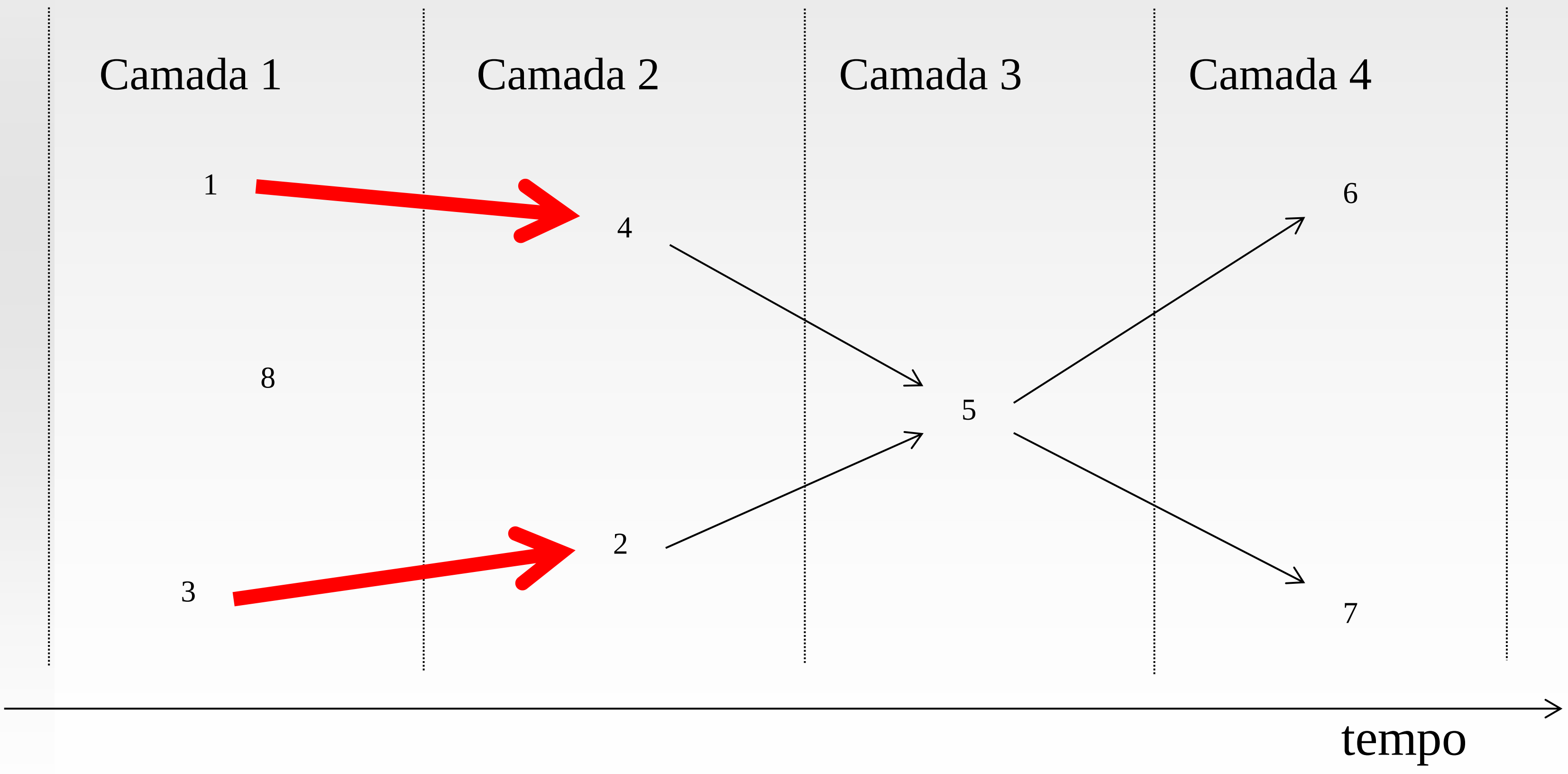
- O algoritmo proposto está dividido em duas fases:
  - construtiva e
  - melhoramento.

Construção da Solução Inicial

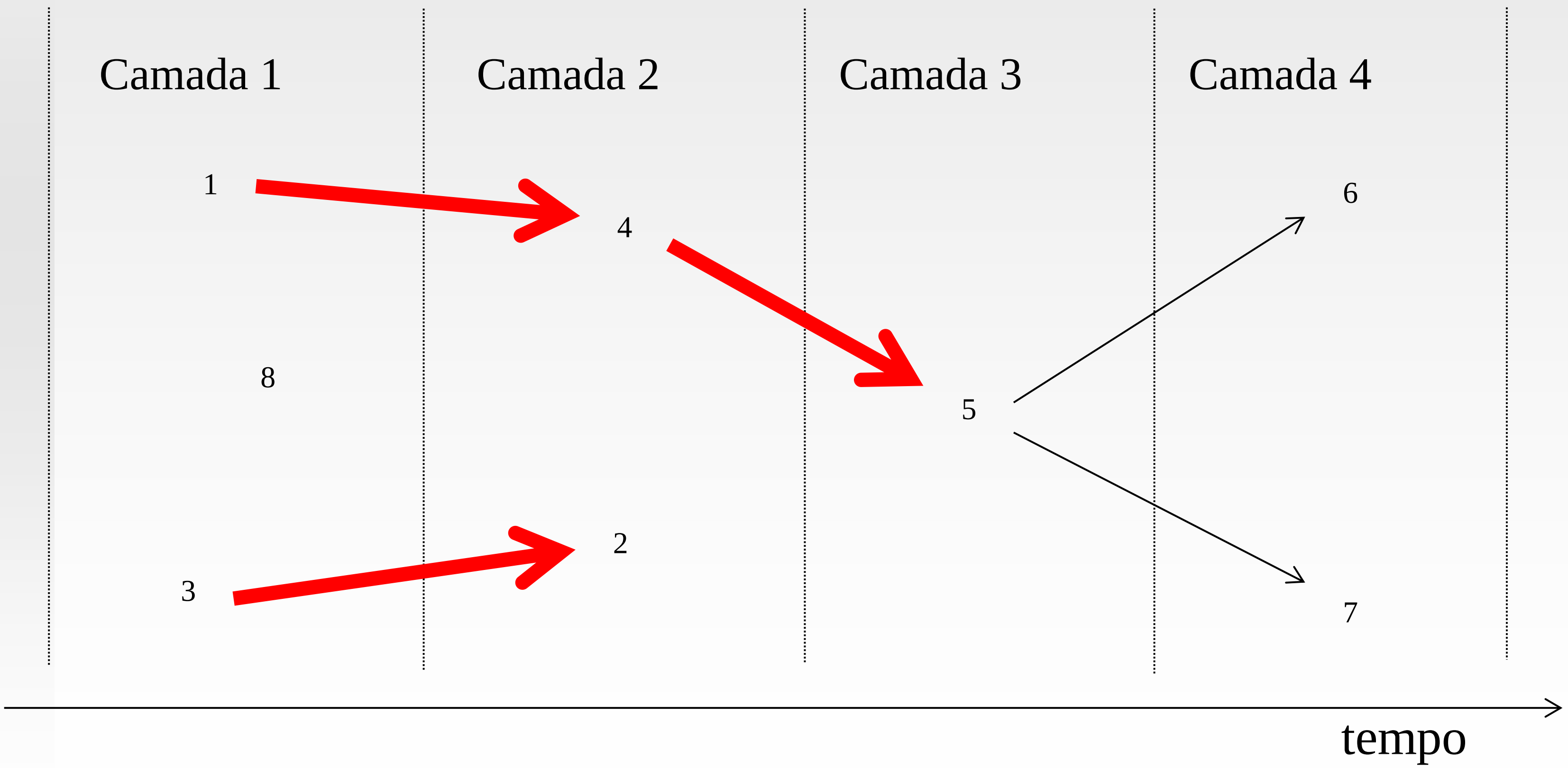
Formação das camadas



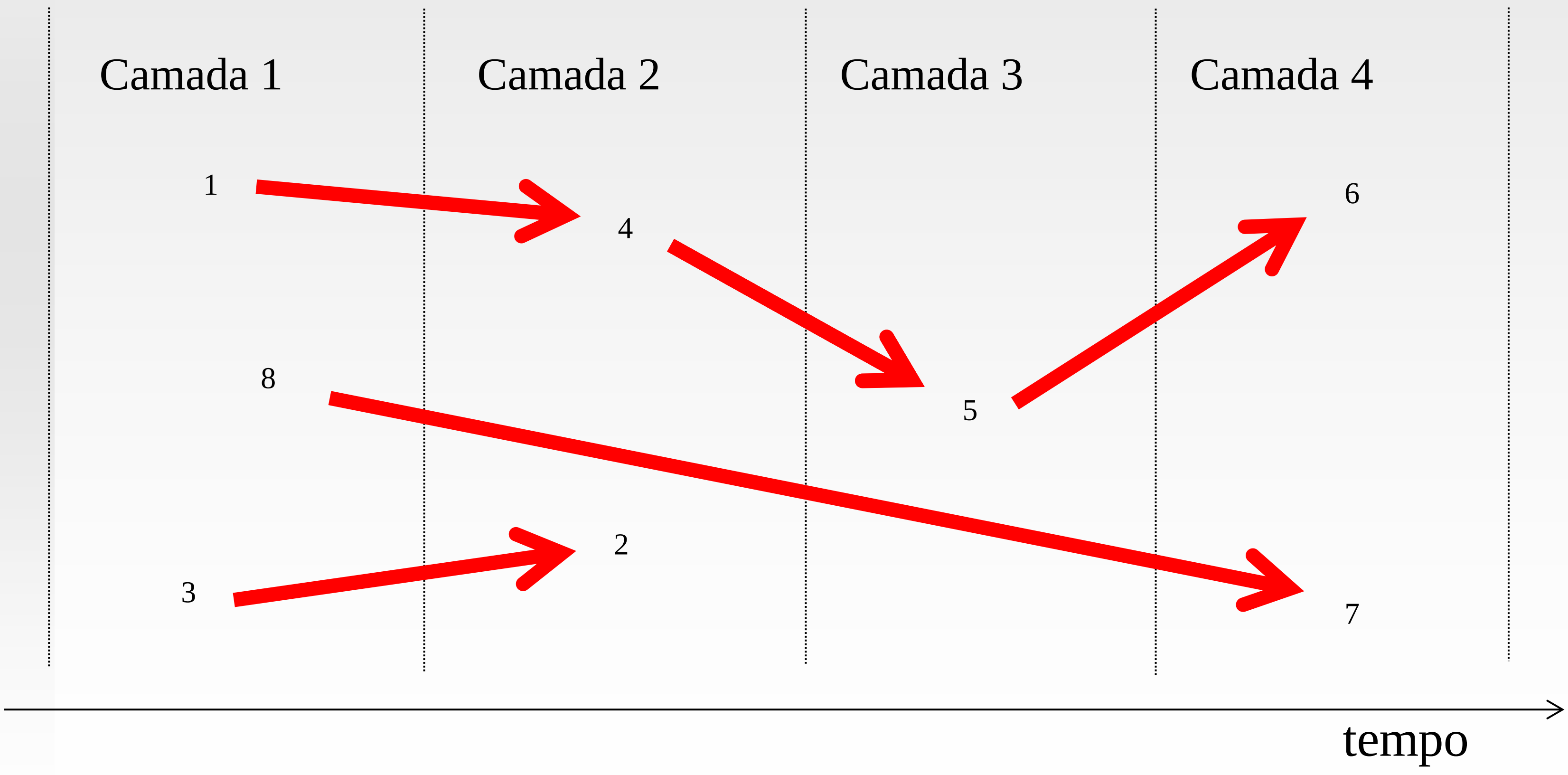
Construção da Solução Inicial



Construção da Solução Inicial

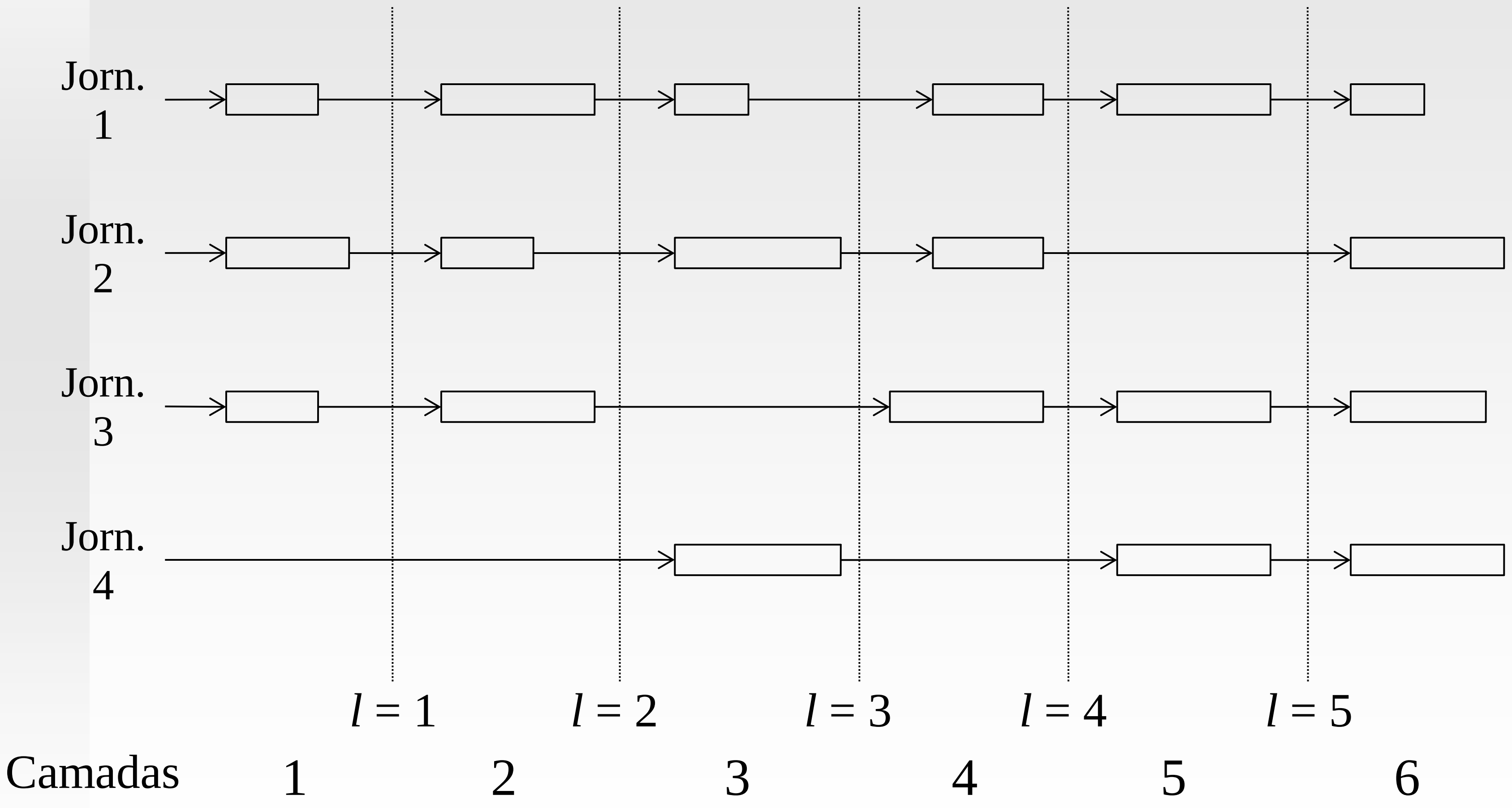


Construção da Solução Inicial



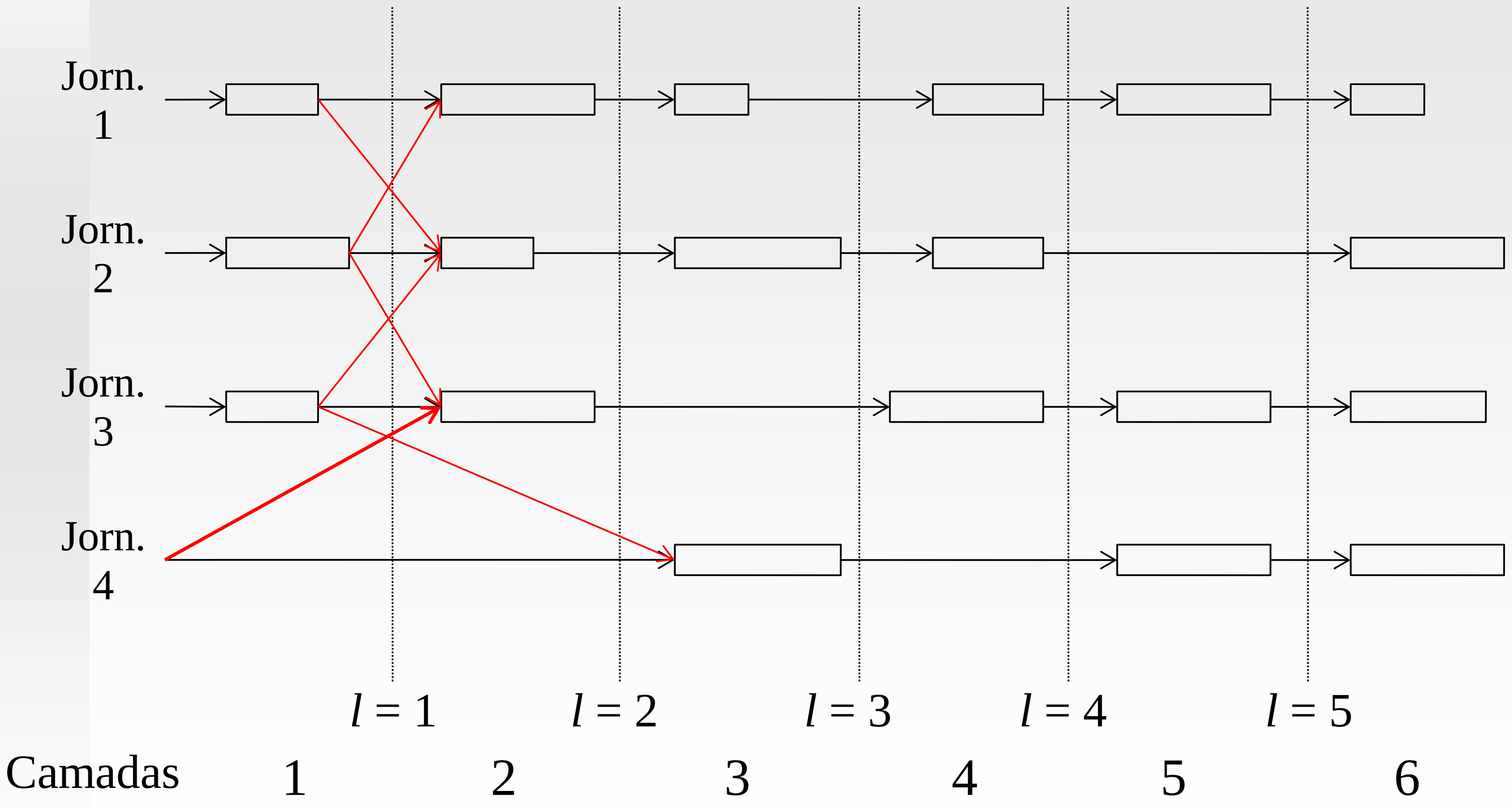
Fase de Melhoramento: **M1**

A solução inicial e os possíveis cortes



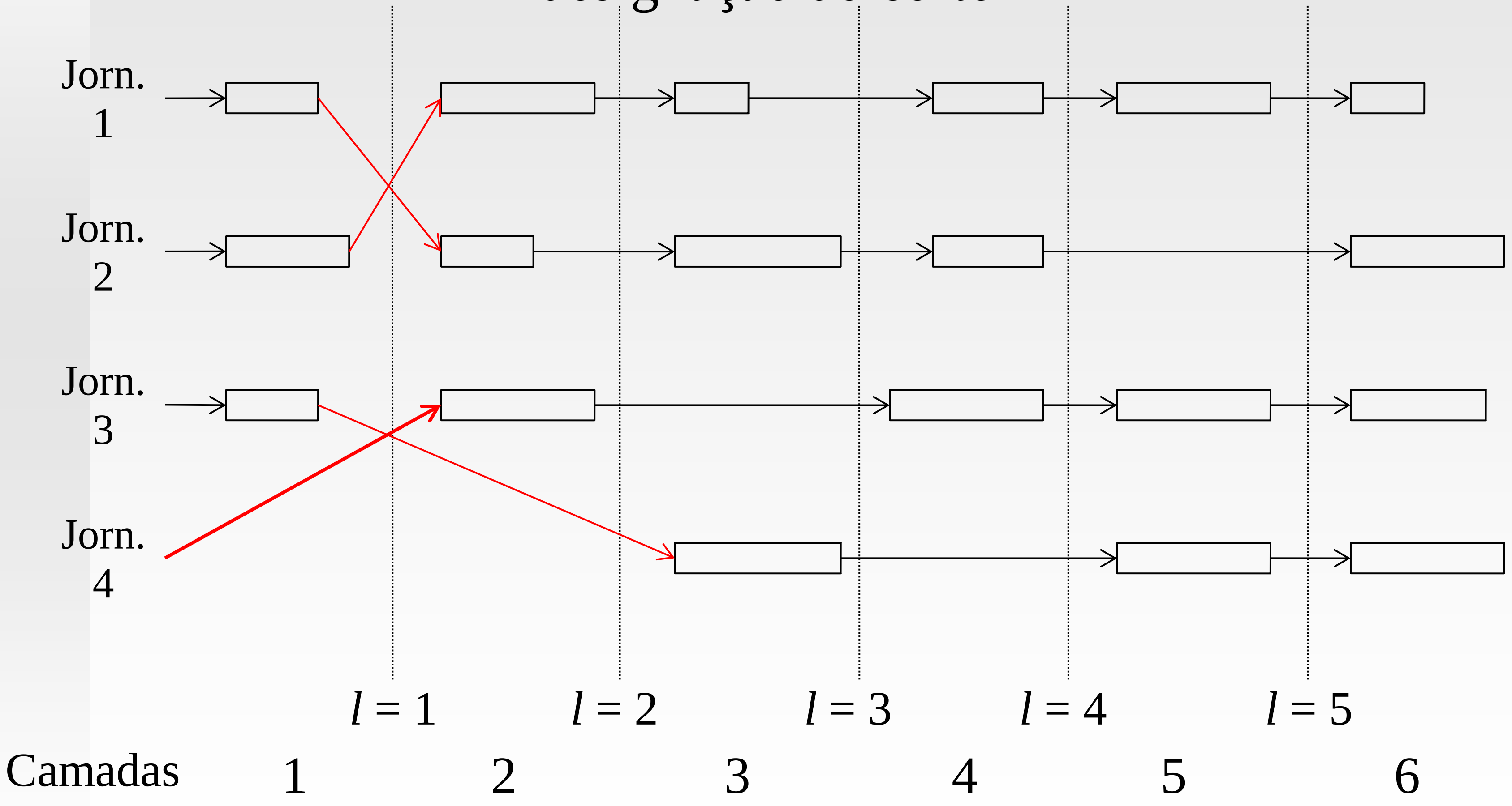
Fase de Melhoramento: M1

Possíveis recombinações de jornadas para o corte 1



Fase de Melhoramento: M1

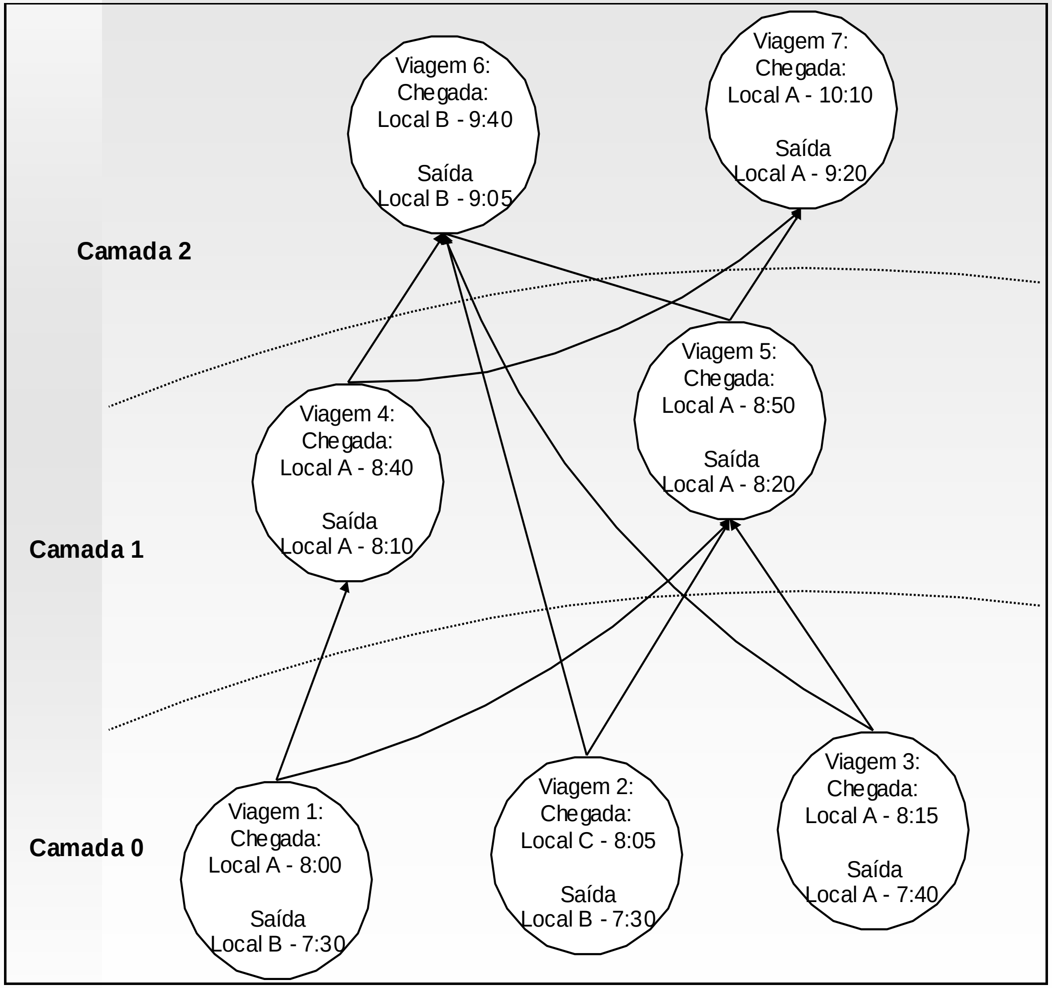
Uma possível solução após a resolução do problema de designação do corte 1



Emparelhamento

- Modelo de Atribuição

Grafo: camadas de viagens



	1	2	3	4	5	6	7
1	0	0	0	1	1	0	0
2	0	0	0	0	1	1	0
3	0	0	0	0	1	1	0
4	0	0	0	0	0	1	1
5	0	0	0	0	0	1	1
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

Emparelhamento: Matriz de Custo

Viagens

Veículos

1º quadrante posição [i, j]: custo de se alocar a viagem v_i ao veículo u_i	**2º quadrante** posição [i, j]: custo de não se alocar nenhuma viagem ao veículo u_{ii}
3º quadrante posição [i, j]: custo de não se realizar a viagem v_i	**4º quadrante** posição [i, j]: valor nulo

3° Caso:

Escalonamento de Maquinistas em uma Empresa de Transporte Ferroviário de Carga.

Característica do problema

(Companhia Vale do Rio Doce)

- Um conjunto de *n* atividades que se repetem todos os dias
- Tipos de atividades: Prontidão, Manobra, Help, Trem Cargueiro, Trem de Minério, ...
- Duração das atividades: 30 horas, 6 horas.
- Um conjunto de restrições legais, contratuais e sindicais.

Objetivo: Produzir escalas mensais para os condutores de maneira que a carga de trabalho fique distribuída equitativamente.

SOLUÇÃO PARA O PROBLEMA

- Construção da Escala Cíclica
  - Modelo *Set Covering*
  - Algoritmo Heurístico

- Distribuição das Escalas para os Condutores
  - Algoritmo Heurístico
    - Heurística 2-opt
      - Problema de Atribuição com Gargalo
        - Função Utilidade - Pref. Declarada

Construção da Escala Cíclica

DIA 1	DIA 2	DIA 3	DIA 4	DIA 5	DIA 6	DIA 7
a_1	a_2		a_3	a_4	X	a_5
a_6		a_7	a_8	X	a_9	a_{10}
a_{11}		a_{12}	X	a_{13}		a_{14}
a_{15}	a_{16}	X	a_{17}		a_{18}	a_{19}
a_{20}	X	a_{21}	a_{22}		a_{23}	
X	a_{24}		a_{25}	a_{26}	a_{27}	X

Conjunto das atividades $A=\{a_1, a_2, ..., a_{27}\}$

Programação P_j = um conjunto ordenado de atividades entre folgas

Escala

1- Modelo *Set Covering* – Cobertura de Conjunto

$$\text{Min} \sum_{J=1}^{n} c_j x_j$$

sujeito à

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1, \quad i = 1, \ldots, n$$

$$x_j \geq 0, \; j = 1, \ldots, m$$

onde:

c_j = custo da programação j;

x_j=1 se a programação j for escolhida e 0 caso contrário;

a_{ij}= 1 se a atividade i for executada plea programação j, e 0 caso contrário

m = o número de escalas alternativas possíveis;

n = número de atividades.

Objetivo: obter um conjunto de programações de custo mínimo.

Exemplo:

Minimizar $(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7)$

Sujeito à

$$\begin{vmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{vmatrix} \geq \begin{vmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{vmatrix}$$

2- Algoritmo Heurístico

1°. Estimar o número de programações:

$$s \geq \left(\frac{1}{1440} \frac{z}{\alpha} \right)$$

2°. construir *s* programação.

Distribuição das Escalas para os Condutores

U(P_j) = utilidade da programação *j*
S^* = escala ótima
U(S^*)= U(P_1)+ U(P_2)+ ... + U(P_s)

•horizonte de planejamento = 30 dias

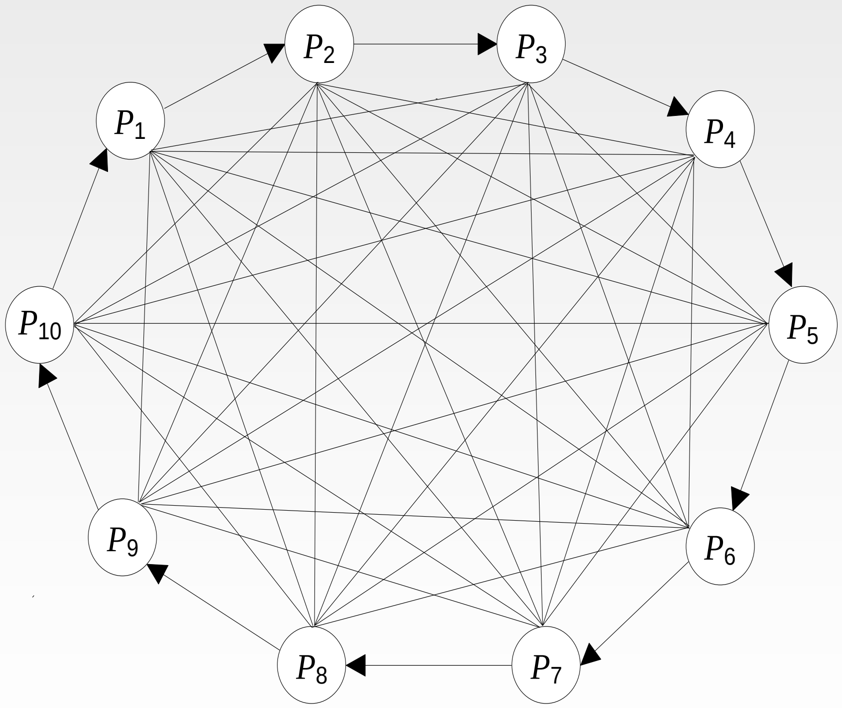
DIA 1	DIA 2	DIA 3	DIA 4	DIA 5	DIA 6	DIA 7
a_1	a_2		a_3	a_4	X	a_5
a_6		a_7	a_8	X	a_9	a_{10}
a_{11}		a_{12}	X	a_{13}		a_{14}
a_{15}	a_{16}	X	a_{17}		a_{18}	a_{19}
a_{20}	X	a_{21}	a_{22}		a_{23}	
X	a_{24}		a_{25}	a_{26}	a_{27}	X

$S_1=\{a_1, a_2, ..., X, a_{21}\}$ U(S_1)
$S_2=\{a_2, a_3, ..., X, a_{21}, a'_{22}\}$ U(S_2)
...
$S_{42}=\{X, a_1, a_2, ..., a_{20}, X\}$ U(S_{42})

Problema de atribuição com gargalo

onde c_{ij}=(-1)(U(H_i) + U(S_j)) se o condutor *i* pode continuar com a escala truncada S_j , c_{ij}= ∞ caso contrário.

Ciclo de Programações



Ajuste da Função Utilidade - Preferência Declarada

1- Identificação dos atributos e seus níveis

HS - total de horas de trabalho na programação

NT - mix de atividades por tipo na programação;

HN -percentual de horas noturnas sobre o total de horas de trabalho;

NP - medida de progressividade;

FG- dia da semana em que a folga é cumprida.

2- Projeto do experimento (12 X 8 = 96 cartões)

3- Elaboração dos cartões

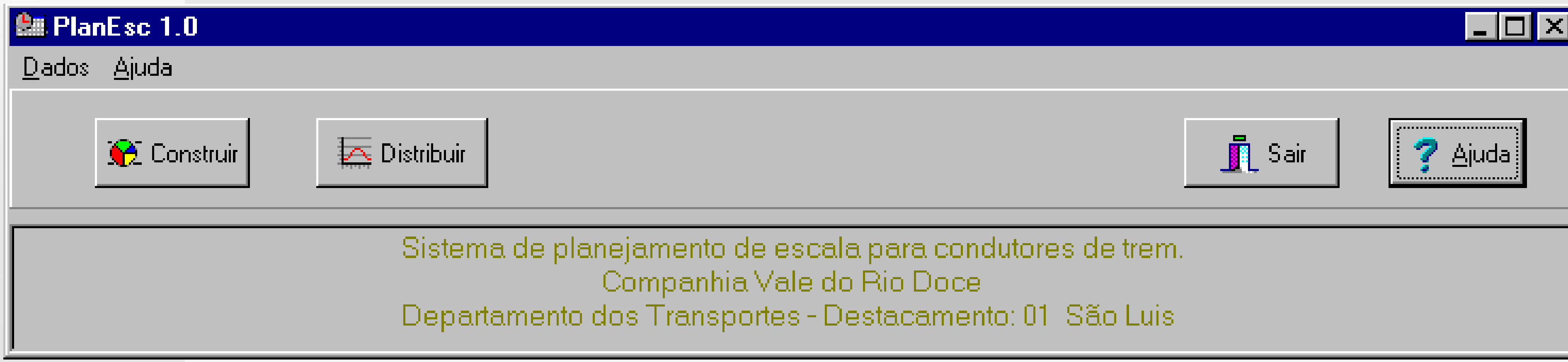
4- Realização das entrevistas

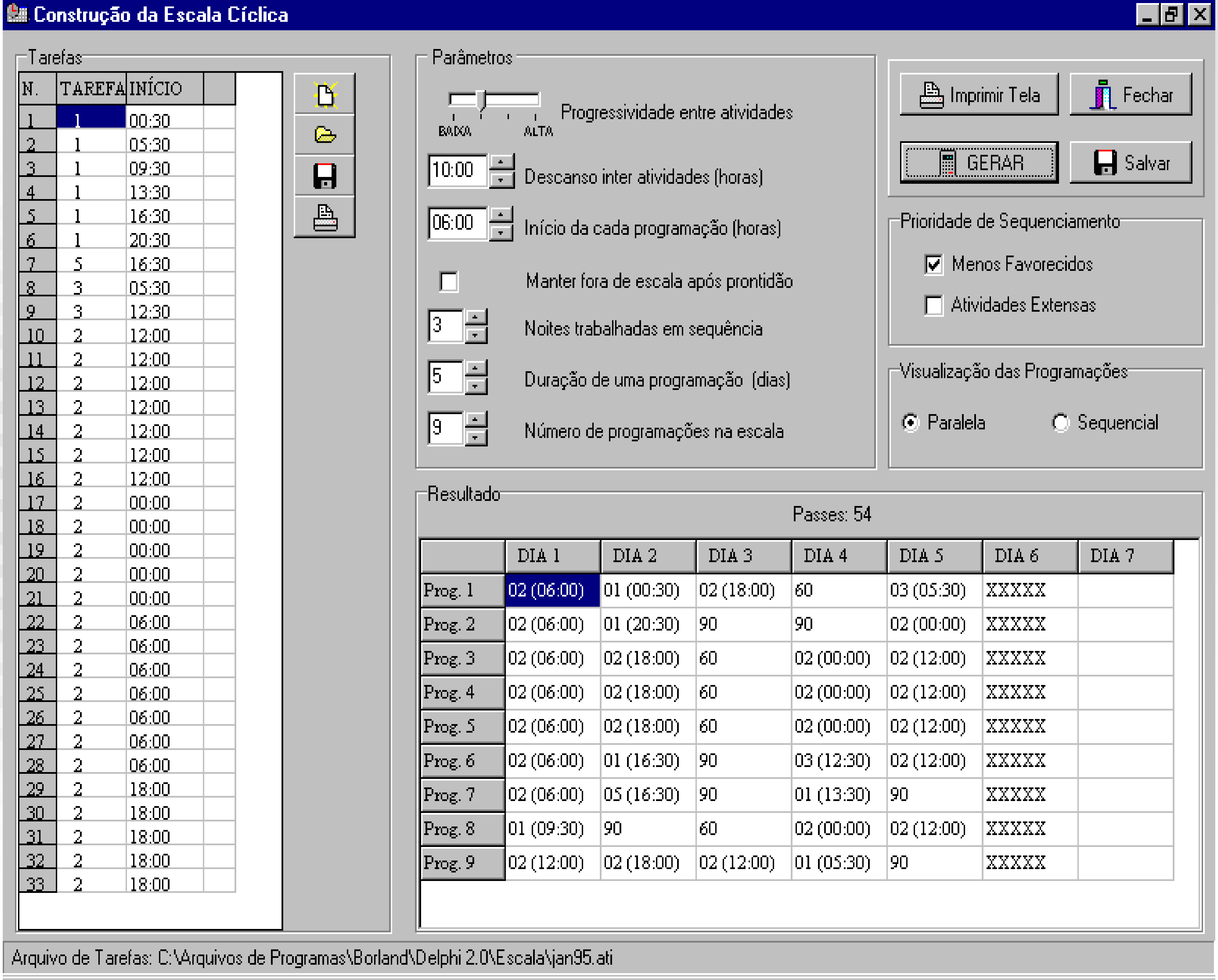
5- Calibração dos parâmetros (Modelo Logit Multinomial)

6- Função Utilidade:

$$U = - 5.42\ T_1 - 5.92\ T_5 - 2.15\ T_2 - 3.04\ T_3 + 7.78\ FG - 12.24\ HS/10 - 1.78\ HN$$

O Sistema Computacional





N.	TAREFA	INÍCIO
1	1	00:30
2	1	05:30
3	1	09:30
4	1	13:30
5	1	16:30
6	1	20:30
7	5	16:30
8	3	05:30
9	3	12:30
10	2	12:00
11	2	12:00
12	2	12:00
13	2	12:00
14	2	12:00
15	2	12:00
16	2	12:00
17	2	00:00
18	2	00:00
19	2	00:00
20	2	00:00
21	2	00:00
22	2	06:00
23	2	06:00
24	2	06:00
25	2	06:00
26	2	06:00
27	2	06:00
28	2	06:00
29	2	18:00
30	2	18:00
31	2	18:00
32	2	18:00
33	2	18:00

	DIA 1	DIA 2	DIA 3	DIA 4	DIA 5	DIA 6	DIA 7
Prog. 1	02 (06:00)	01 (00:30)	02 (18:00)	60	03 (05:30)	XXXXX	
Prog. 2	02 (06:00)	01 (20:30)	90	90	02 (00:00)	XXXXX	
Prog. 3	02 (06:00)	02 (18:00)	60	02 (00:00)	02 (12:00)	XXXXX	
Prog. 4	02 (06:00)	02 (18:00)	60	02 (00:00)	02 (12:00)	XXXXX	
Prog. 5	02 (06:00)	02 (18:00)	60	02 (00:00)	02 (12:00)	XXXXX	
Prog. 6	02 (06:00)	01 (16:30)	90	03 (12:30)	02 (12:00)	XXXXX	
Prog. 7	02 (06:00)	05 (16:30)	90	01 (13:30)	90	XXXXX	
Prog. 8	01 (09:30)	90	60	02 (00:00)	02 (12:00)	XXXXX	
Prog. 9	02 (12:00)	02 (18:00)	02 (12:00)	01 (05:30)	90	XXXXX	

Distribuição da Escalas

Condutores | Escala Cíclica | Distribuição

Fechar

Banco de Dados

65 Registros — 54 Disponíveis

Código	Nome do Condutor	Nasc.	Carga Acum.	Disp.
1	Adailton Silva Soares	01/02/45	1,94	SIM
42	Alvaro J. N. Lindoso	07/05/56	2,3631	SIM
39	André Caciona F. Filho	23/02/41	2,2633	SIM
14	Antônio Alves Pereira	01/02/42	2,3143	SIM
20	Antonio Jose S. da Silva	07/08/45	2,4786	SIM
2	Antonio Lopes Rosado	06/04/53	3,1929	SIM
60	Antônio Luiz de O. Lima	15/04/48	3,1042	NÃO
44	Benedito Macedo Lima	19/10/46	2,4472	SIM

Editar | Inserir | Apagar

Relatório

Código	Tipo	Início	Semana	Data
20	90	00:00:00	Quinta	01/05/97
20	2	12:00:00	Sexta	02/05/97
20	2	06:00:00	Sábado	03/05/97
20	2	06:00:00	Domingo	04/05/97
20	50	00:00:00	Segunda	05/05/97
20	1	06:00:00	Terça	06/05/97
20	90	00:00:00	Quarta	07/05/97
20	2	23:00:00	Quinta	08/05/97
20	2	23:00:00	Sexta	09/05/97
20	2	06:00:00	Sábado	10/05/97
20	50	00:00:00	Domingo	11/05/97
20	3	05:30:00	Segunda	12/05/97
20	60	00:00:00	Terça	13/05/97

Editar | Inserir | Apagar

Estatística do Condutor

Tipos de Atividades

Tipo	Descrição
1	Viagem Minério
2	Manobra
3	Prontidão
5	Viagem Cargueiro
9	Help
50	Folga
60	Fora de Escala
90	Continuação

Distribuição da Escalas

Condutores | Escala Cíclica | Distribuição

Fechar

Parâmetros para junções

Progressividade entre atividades

BAIXA ALTA

Manter fora de escala após prontidão.

10:00 Descanso inter atividades (horas)

06:00 Início da cada programação (horas)

3 Noites trabalhadas em sequência.

Escala Cíclica

Abrir Aquivo: Jan95.esc Passes: 54

DIA 1	DIA 2	DIA 3	DIA 4	DIA 5	DIA 6	DIA 7
02 (06:00)	01 (00:30)	02 (18:00)	60	03 (05:30)	XXXXX	02 (06:00)
01 (20:30)	90	90	02 (00:00)	XXXXX	02 (06:00)	02 (18:00)
60	02 (00:00)	02 (12:00)	XXXXX	02 (06:00)	02 (18:00)	60
02 (00:00)	02 (12:00)	XXXXX	02 (06:00)	02 (18:00)	60	02 (00:00)
02 (12:00)	XXXXX	02 (06:00)	01 (16:30)	90	03 (12:30)	02 (12:00)
XXXXX	02 (06:00)	05 (16:30)	90	01 (13:30)	90	XXXXX
01 (09:30)	90	60	02 (00:00)	02 (12:00)	XXXXX	02 (12:00)
02 (18:00)	02 (12:00)	01 (05:30)	90	XXXXX		

Período de Planejamento

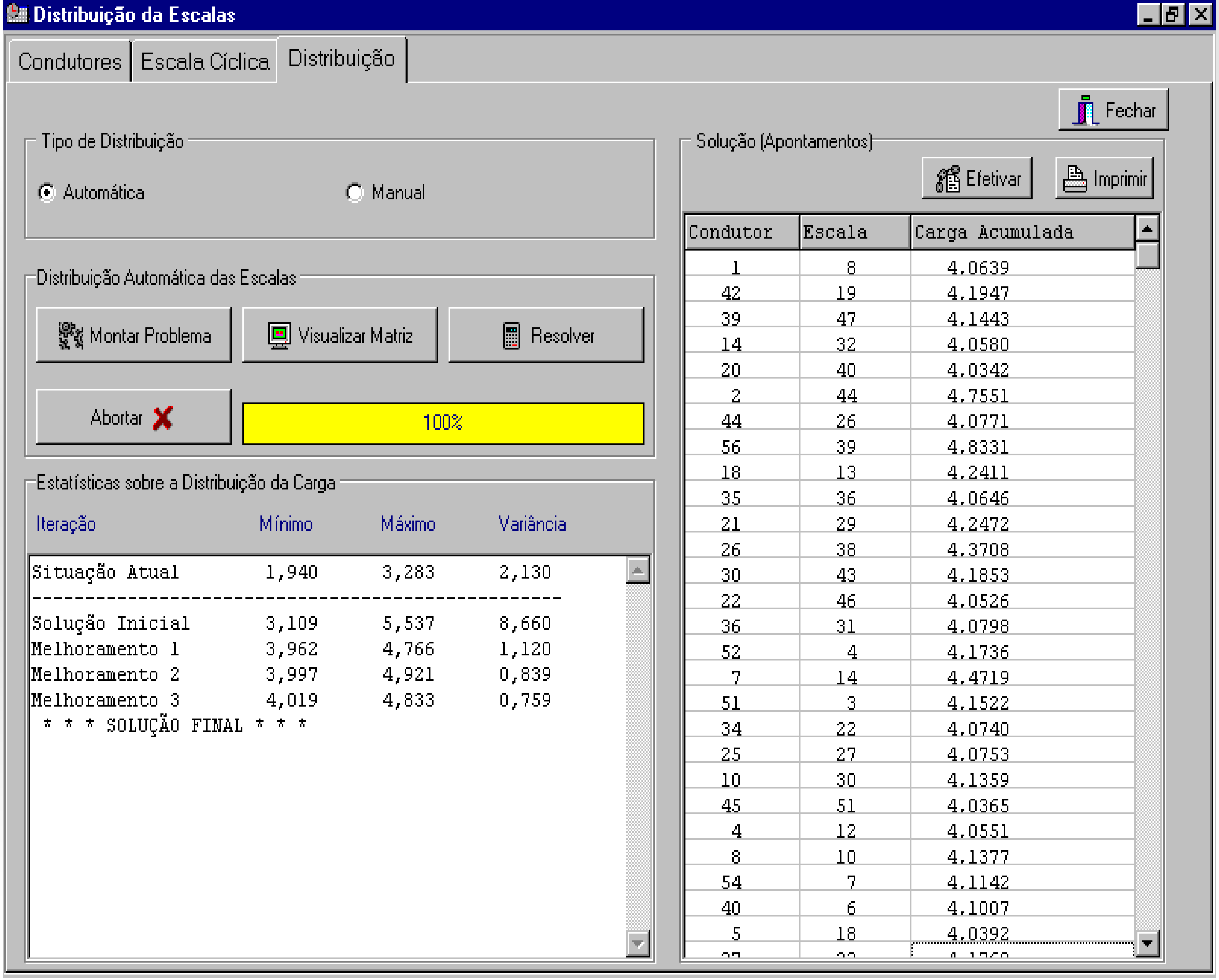
Início (Dia/Mês/Ano) 01/06/97

Horizonte (Dias) 30

Escala Truncada

Número 1 Utilidade 2,3440

Domingo	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
02 (06:00)	01 (00:30)	02 (18:00)	60	03 (05:30)	XXXXX	02 (06:00)
01 (20:30)	90	90	02 (00:00)	XXXXX	02 (06:00)	02 (18:00)
60	02 (00:00)	02 (12:00)	XXXXX	02 (06:00)	02 (18:00)	60
02 (00:00)	02 (12:00)	XXXXX	02 (06:00)	02 (18:00)	60	02 (00:00)
02 (12:00)	XXXXX					


Distribuição da Escalas
Condutores
Escala Cíclica
Distribuição
Fechar
Tipo de Distribuição
Automática
Manual
Distribuição Automática das Escalas
Montar Problema
Visualizar Matriz
Resolver
Abortar
100%
Estatísticas sobre a Distribuição da Carga
Iteração Mínimo Máximo Variância
Situação Atual 1,940 3,283 2,130
Solução Inicial 3,109 5,537 8,660
Melhoramento 1 3,962 4,766 1,120
Melhoramento 2 3,997 4,921 0,839
Melhoramento 3 4,019 4,833 0,759
* * * SOLUÇÃO FINAL * * *
Solução (Apontamentos)
Efetivar
Imprimir
Condutor Escala Carga Acumulada
1 8 4.0639
42 19 4.1947
39 47 4.1443
14 32 4.0580
20 40 4.0342
2 44 4.7551
44 26 4.0771
56 39 4.8331
18 13 4.2411
35 36 4.0646
21 29 4.2472
26 38 4.3708
30 43 4.1853
22 46 4.0526
36 31 4.0798
52 4 4.1736
7 14 4.4719
51 3 4.1522
34 22 4.0740
25 27 4.0753
10 30 4.1359
45 51 4.0365
4 12 4.0551
8 10 4.1377
54 7 4.1142
40 6 4.1007
5 18 4.0392

Conclusões

- Mais agilidade nas tomadas de decisões;
- Maior satisfação dos maquinistas;
- Redução de custos ativos e passivos.

Considerações Finais

- Veremos outras aplicações ao longo do curso.