Algoritmos de Aproximação

Segundo Semestre de 2012

Problema dos k-centros

Dados: um grafo completo $G = (V, E)$, um inteiro $k > 0$ e distâncias d_{ij} para cada i e j em V tais que $d_{ii} = 0$ para todo i, $d_{ij} = d_{ji}$ para todo i e j, e $d_{ij} \leq d_{i\ell} + d_{\ell j}$.

Problema dos k-centros

Dados: um grafo completo $G = (V, E)$, um inteiro $k > 0$ e distâncias d_{ij} para cada i e j em V tais que $d_{ii} = 0$ para todo i, $d_{ij} = d_{ji}$ para todo i e j, e $d_{ij} \leq d_{i\ell} + d_{\ell j}$.

S: conjunto não vazio de vértices

$d(i, S) := \min\{d(i, j) : j \in S\}$

raio de S: $\max\{d(i, S) : i \in V\}$

Problema dos k-centros

Dados: um grafo completo $G = (V, E)$, um inteiro $k > 0$ e distâncias d_{ij} para cada i e j em V tais que $d_{ii} = 0$ para todo i, $d_{ij} = d_{ji}$ para todo i e j, e $d_{ij} \leq d_{i\ell} + d_{\ell j}$.

S: conjunto não vazio de vértices

$d(i, S) := \min\{d(i, j) : j \in S\}$

raio de S: $\max\{d(i, S) : i \in V\}$

Objetivo: encontrar um conjunto S com k vértices de V com raio mínimo.

Problema dos k-centros

Dados: um grafo completo $G = (V, E)$, um inteiro $k > 0$ e distâncias d_{ij} para cada i e j em V tais que $d_{ii} = 0$ para todo i, $d_{ij} = d_{ji}$ para todo i e j, e $d_{ij} \leq d_{i\ell} + d_{\ell j}$.

S: conjunto não vazio de vértices

$d(i, S) := \min\{d(i, j) : j \in S\}$

raio de S: $\max\{d(i, S) : i \in V\}$

Objetivo: encontrar um conjunto S com k vértices de V com raio mínimo.

Vamos mostrar uma 2-aproximação para o problema.

Problema dos k-centros

Vamos supor que $V = \{1, \ldots, n\}$.

GULOSO (n, k, d)

1 $S \leftarrow \{1\}$
2 **enquanto** $|S| < k$ **faça**
3 $\quad j \leftarrow \arg\max\{d(\ell, S) : \ell \in V\}$
4 $\quad S \leftarrow S \cup \{j\}$
5 **devolva** S

Problema dos k-centros

Vamos supor que $V = \{1, \ldots, n\}$.

GULOSO (n, k, d)

```
1   S ← {1}
2   enquanto |S| < k faça
3         j ← arg max{d(ℓ, S) : ℓ ∈ V}
4         S ← S ∪ {j}
5   devolva S
```

Teorema: GULOSO é uma 2-aproximação para o problema dos k-centros.

Prova feita em aula.

Problema dos k-centros

Teorema: Se existe uma α-aproximação para o problema dos k-centros com $\alpha < 2$, então P $=$ NP.

Problema dos k-centros

Teorema: Se existe uma α-aproximação para o problema dos k-centros com $\alpha < 2$, então P $=$ NP.

Prova: Redução do conjunto dominante.

Problema dos k-centros

Teorema: Se existe uma α-aproximação para o problema dos k-centros com $\alpha < 2$, então P $=$ NP.

Prova: Redução do conjunto dominante.

Prova feita em aula.

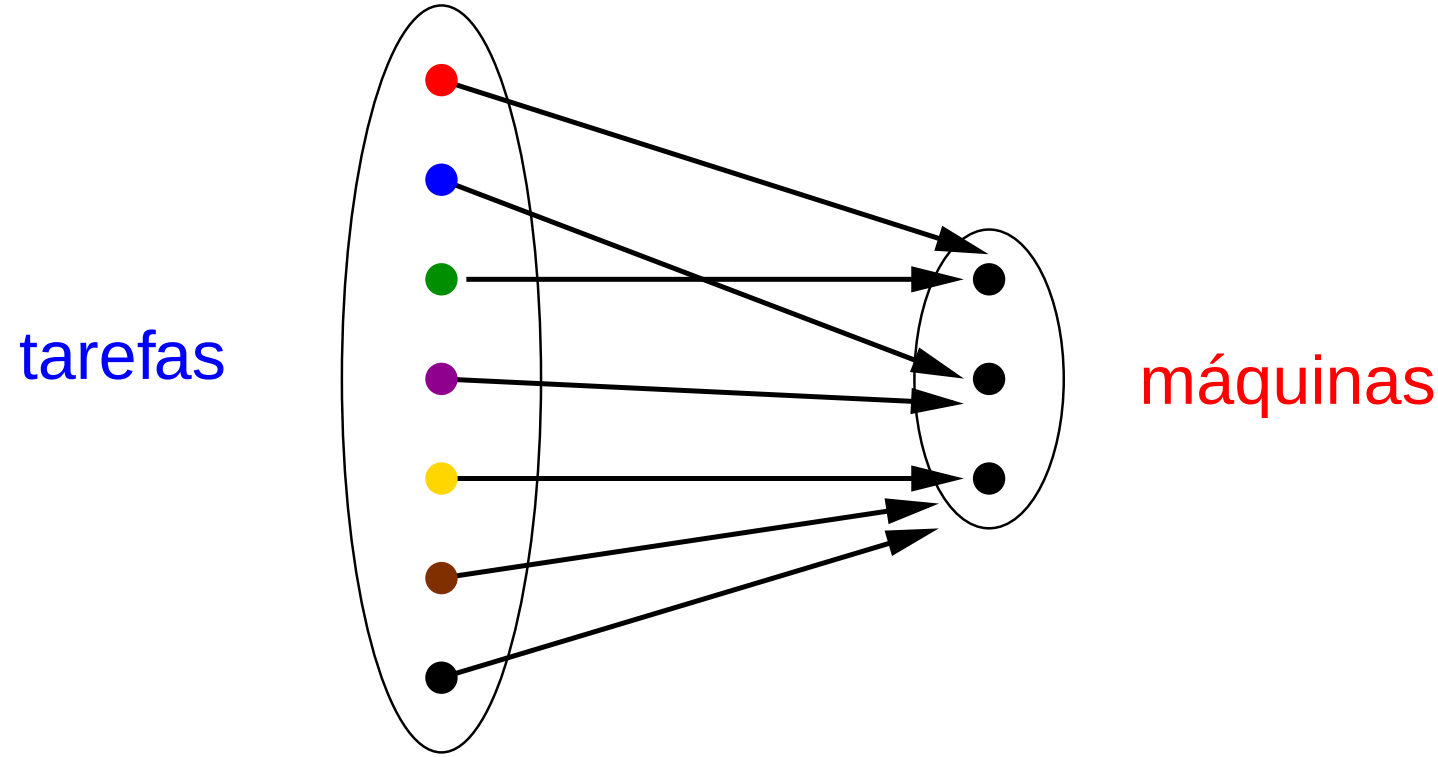
Escalonamento de máquinas idênticas

Dados: m máquinas

n tarefas

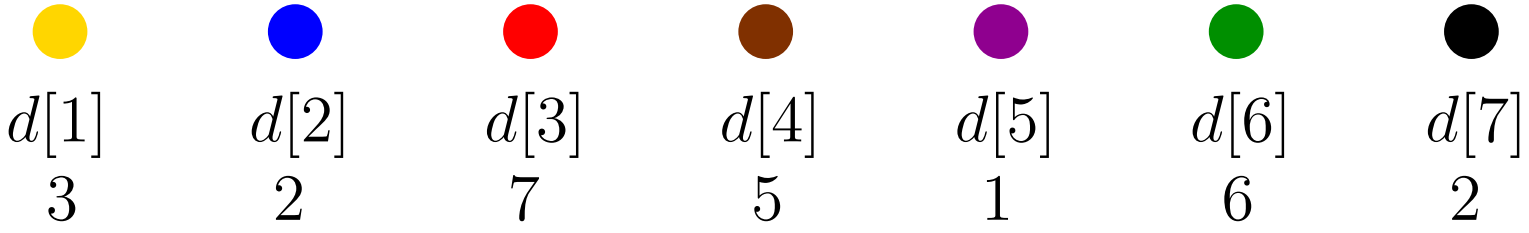
duração $d[i]$ da tarefa i $\quad (i = 1, \ldots, n)$

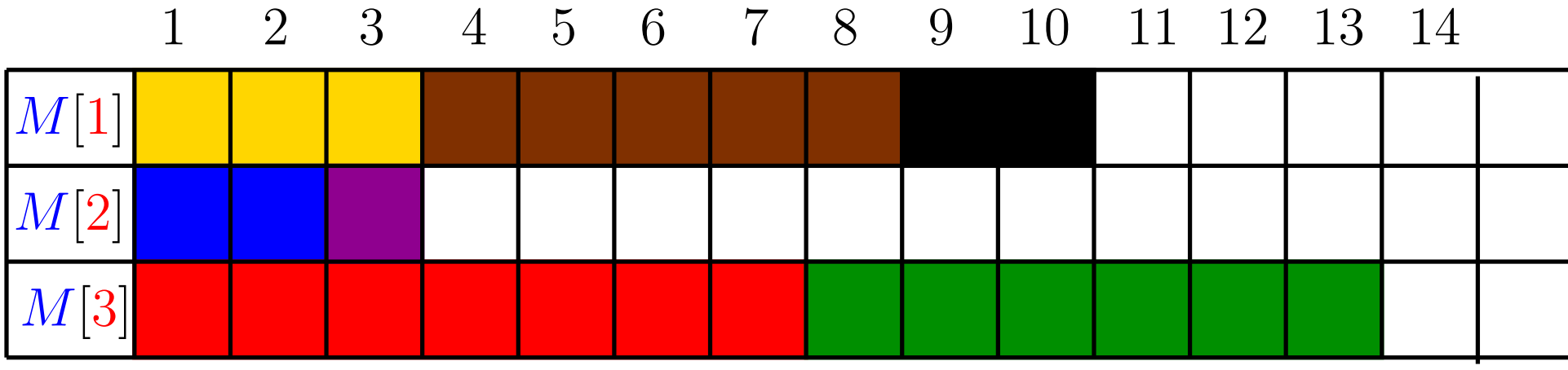
um escalonamento é uma partição $\{M[1], \ldots, M[m]\}$ de $\{1, \ldots, n\}$



Exemplo 1

$m = 3 \quad n = 7$

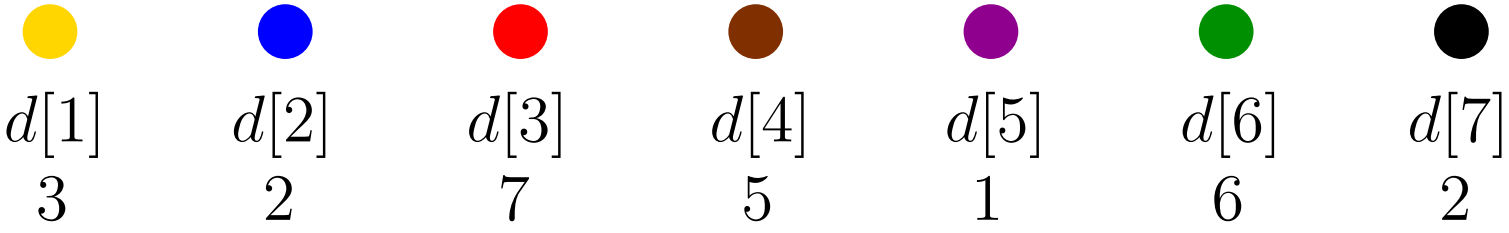


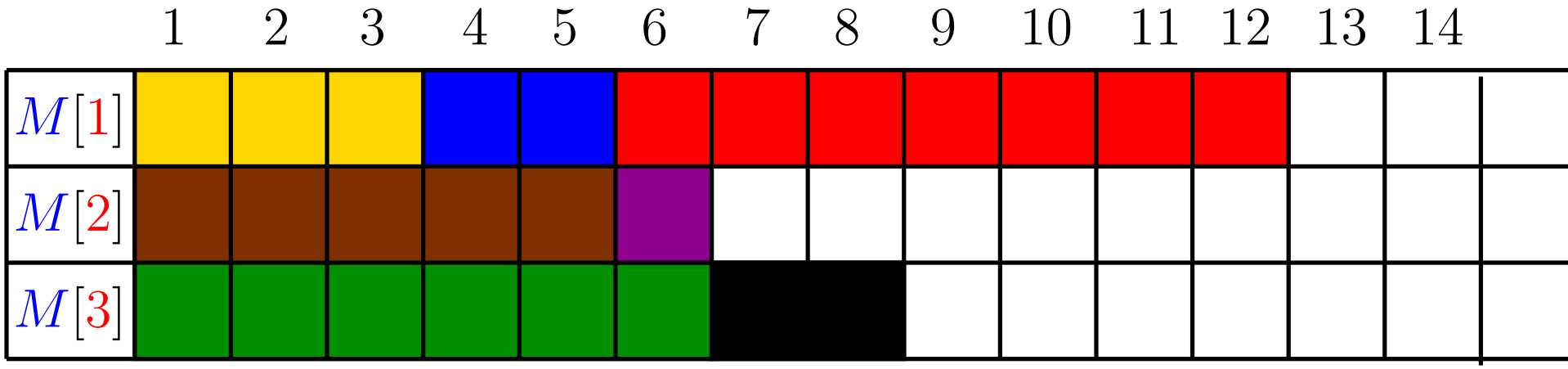


$\{\{1, 4, 7\}, \{2, 5\}, \{3, 6\}\} \Rightarrow$ Tempo de conclusão $= 13$

Exemplo 2

$m = 3 \quad n = 7$

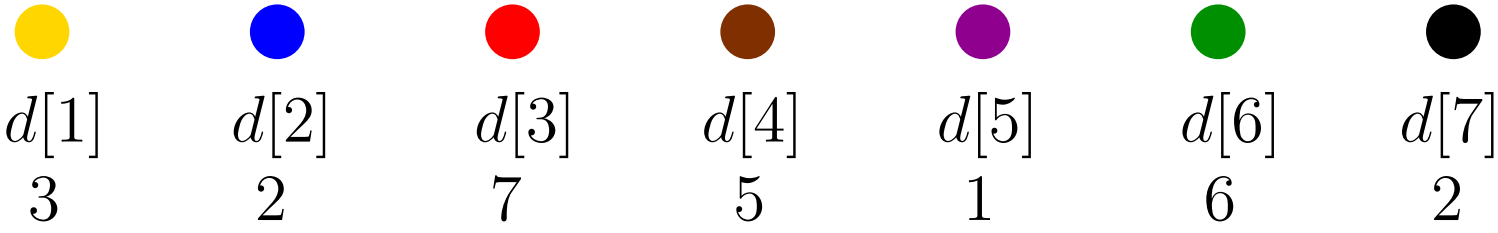


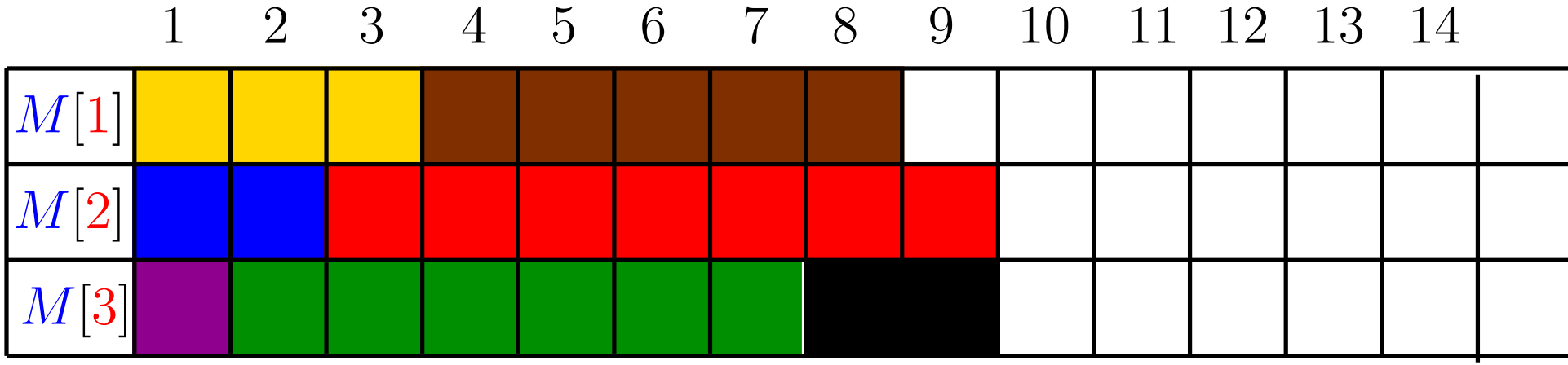


$\{\{1,2,3\},\{4,5\},\{6,7\}\} \Rightarrow$ Tempo de conclusão $= 12$

Problema

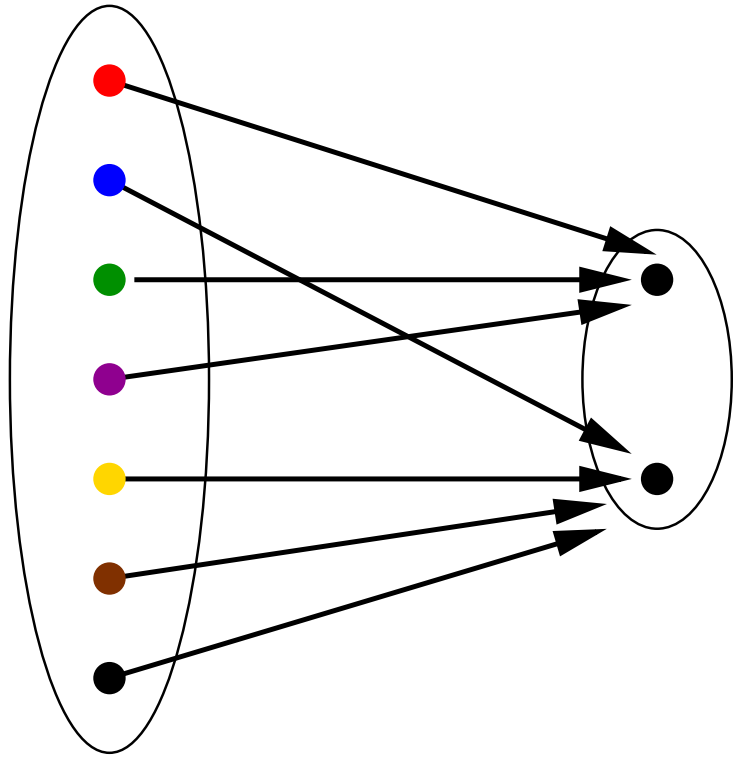
Encontrar um escalonamento com tempo de conclusão mínimo.





$\{\{1,4\},\{2,3\},\{5,6,7\}\} \Rightarrow$ Tempo de conclusão $= 9$

NP-difícil mesmo para $m = 2$



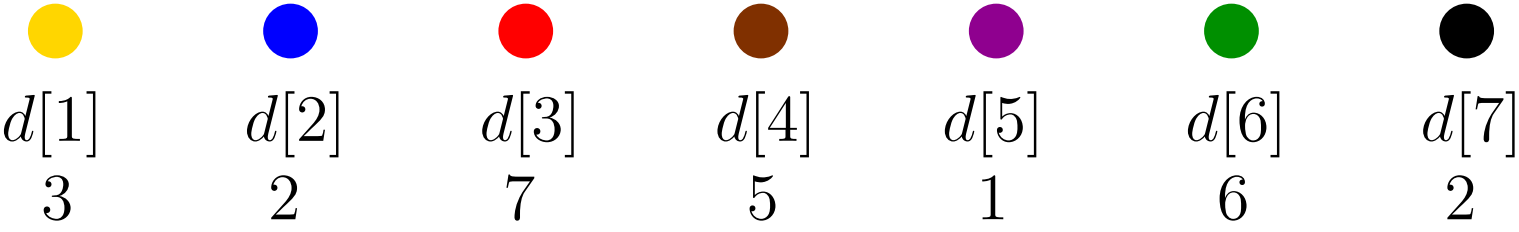
Algoritmo: testa todo $M[1] \subseteq \{1, \ldots, n\}$ e escolhe melhor

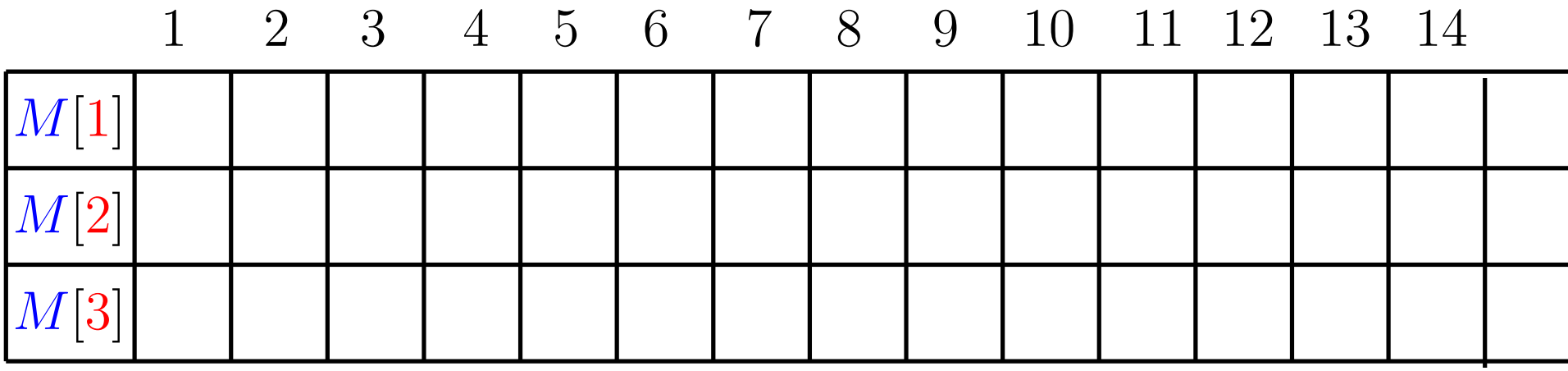
2^n subconjuntos $\Rightarrow$ exponencial

NP-difícil $\Rightarrow$ é improvável que exista algoritmo polinomial que resolva o problema (se exitir, P $=$ NP)

Algoritmo de Graham

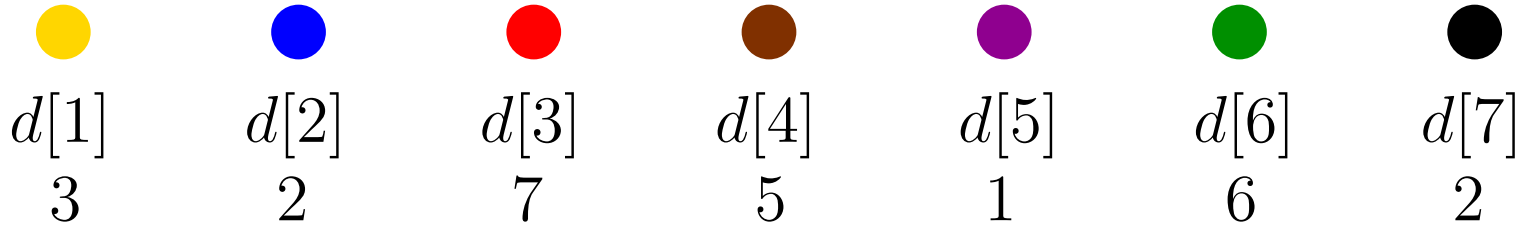
Atribui, uma a uma, cada tarefa à máquina menos ocupada.

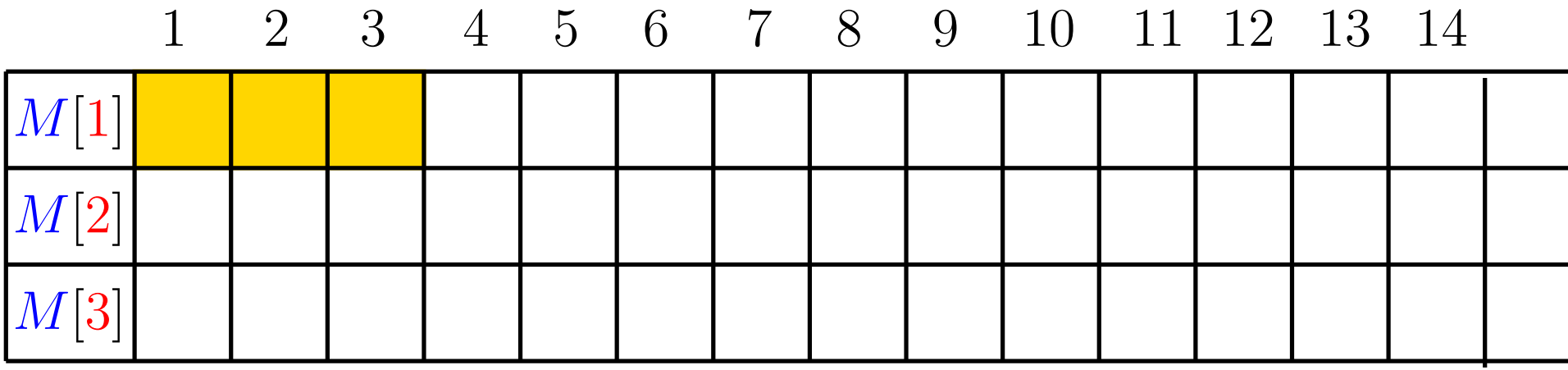




Algoritmo de Graham

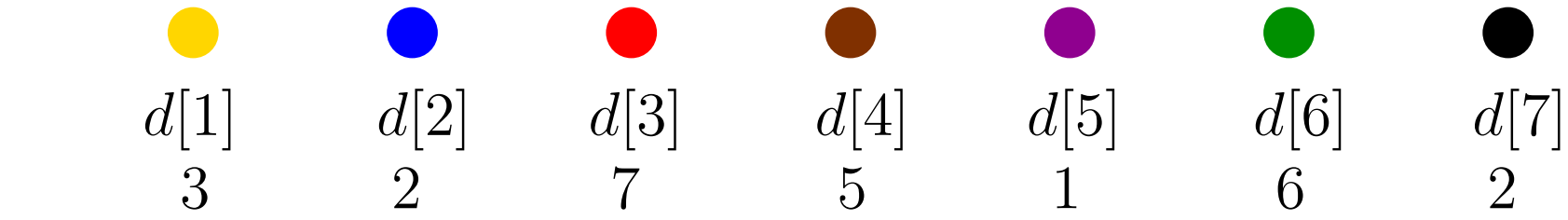
Atribui, uma a uma, cada tarefa à máquina menos ocupada.

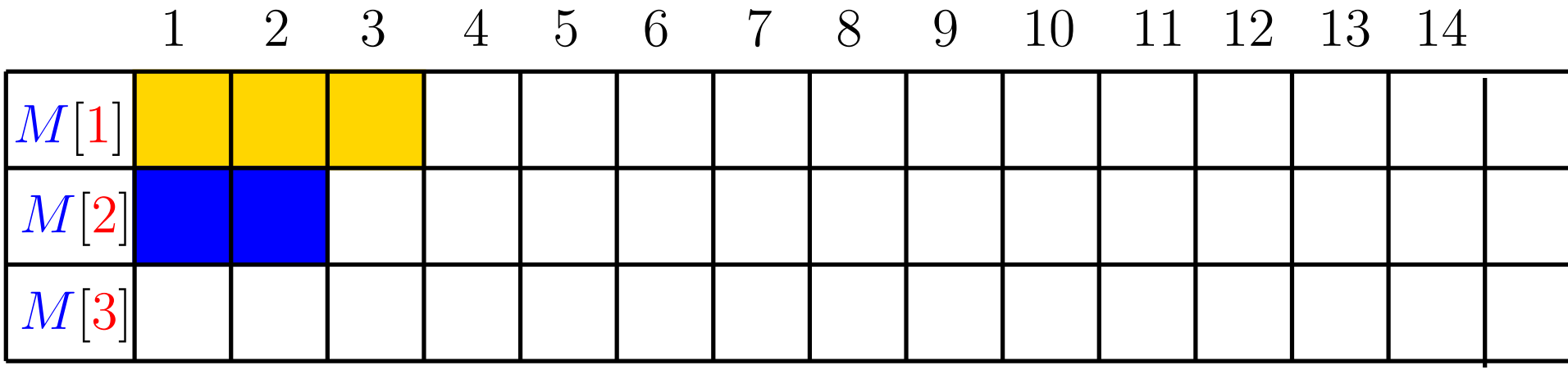




Algoritmo de Graham

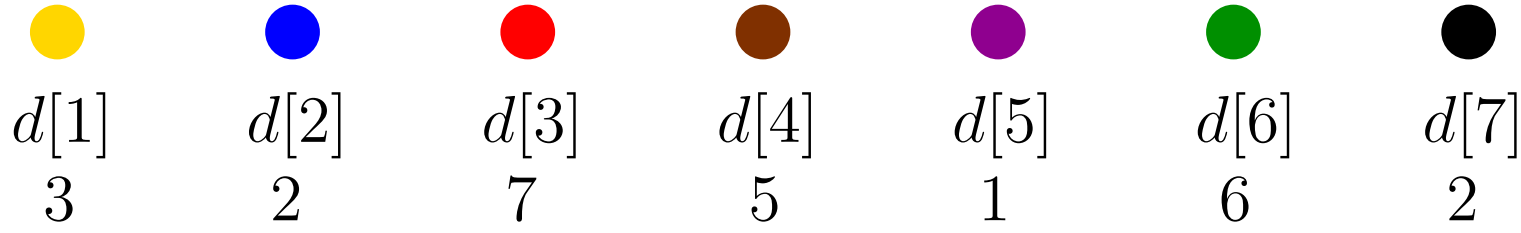
Atribui, uma a uma, cada tarefa à máquina menos ocupada.

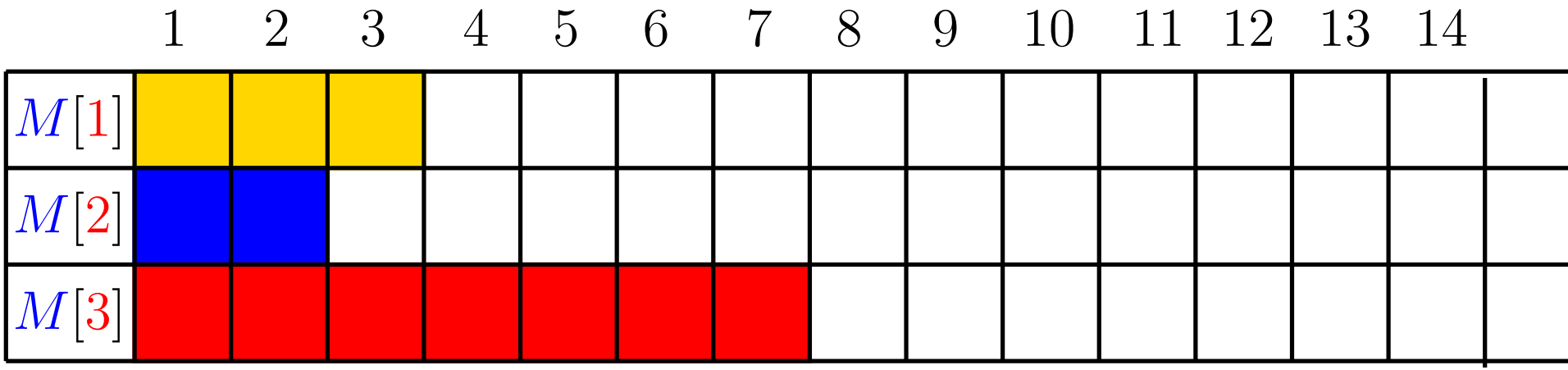




Algoritmo de Graham

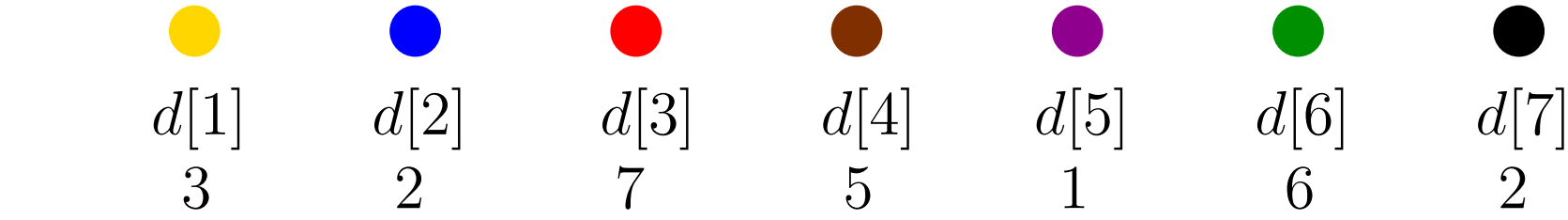
Atribui, uma a uma, cada tarefa à máquina menos ocupada.

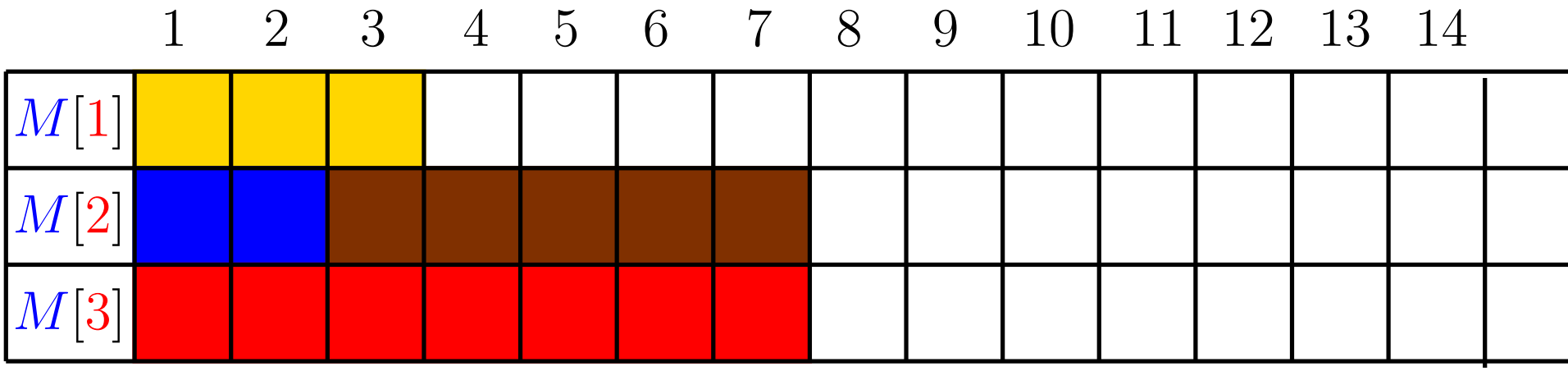




Algoritmo de Graham

Atribui, uma a uma, cada tarefa à máquina menos ocupada.

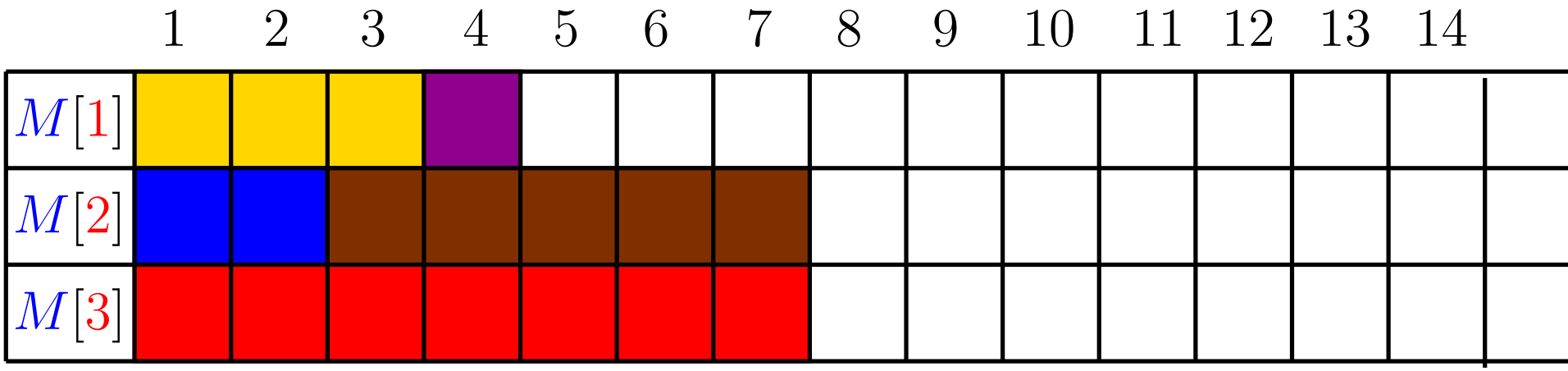




Algoritmo de Graham

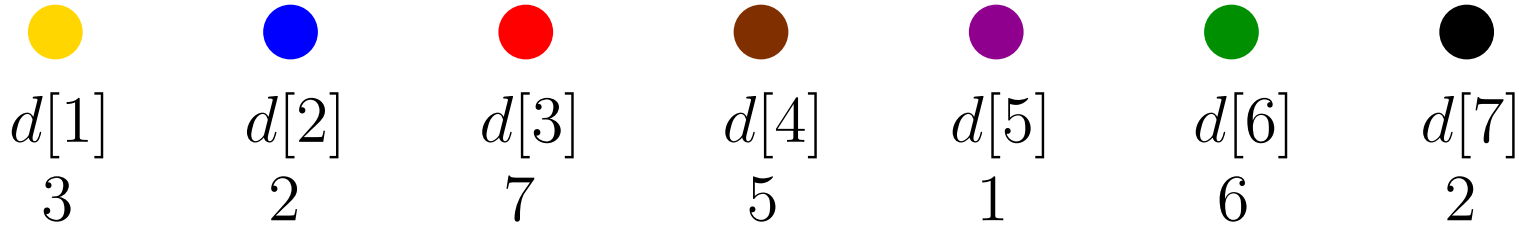
Atribui, uma a uma, cada tarefa à máquina menos ocupada.

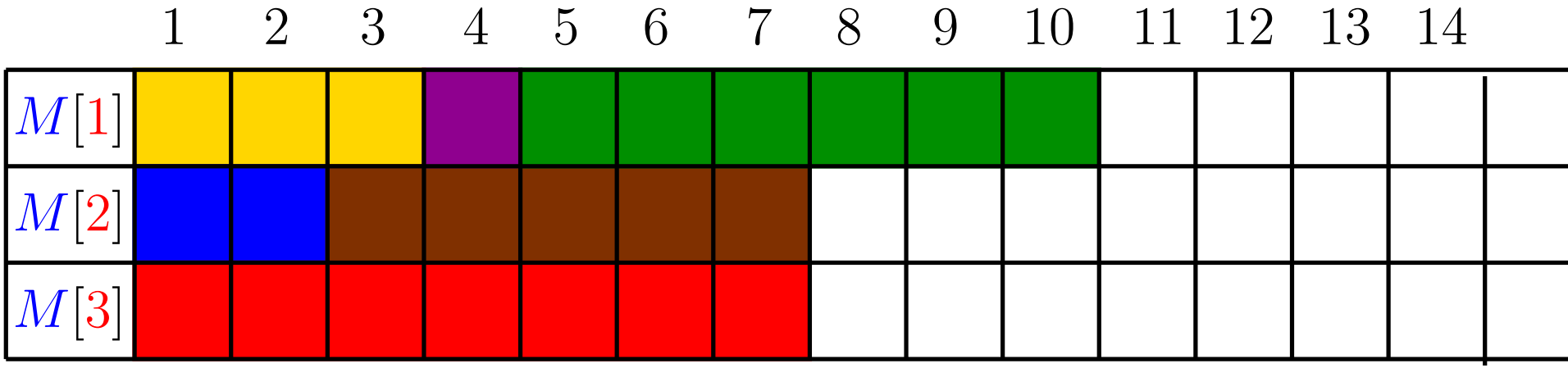
$d[1]$	$d[2]$	$d[3]$	$d[4]$	$d[5]$	$d[6]$	$d[7]$
3	2	7	5	1	6	2



Algoritmo de Graham

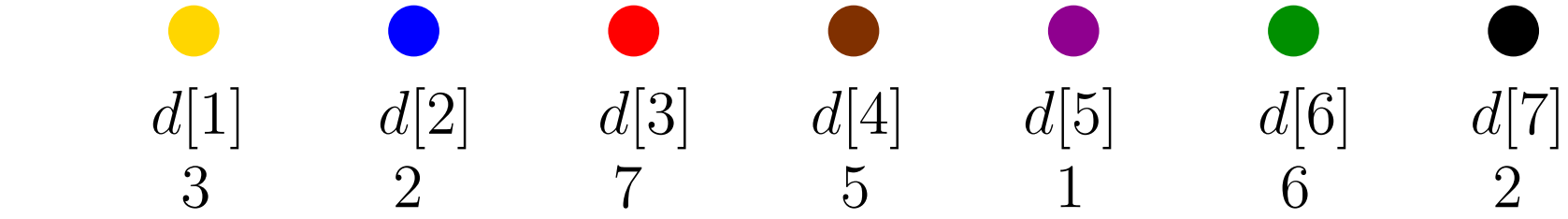
Atribui, uma a uma, cada tarefa à máquina menos ocupada.

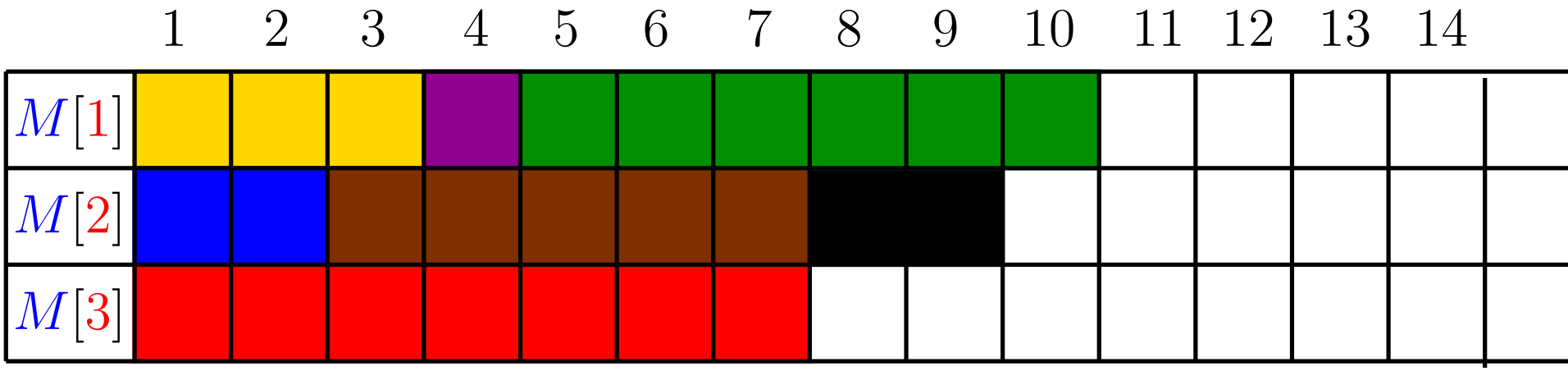




Algoritmo de Graham

Atribui, uma a uma, cada tarefa à máquina menos ocupada.





Escalonamento-Graham

Recebe números inteiros positivos m e n e um vetor $d[1\,..\,n]$ e devolve um escalomento de $\{1, \ldots, n\}$ em m máquinas.

ESCALONAMENTO-GRAHAM (m, n, d)

```
1  para j ← 1 até m faça
2        M[j] ← ∅
3        T[j] ← 0
4  para i ← 1 até n faça
5        seja k tal que T[k] é mínimo
6        M[k] ← M[k] ∪ {i}
7        T[k] ← T[k] + d[i]
8  devolva {M[1], ..., M[m]}
```

Delimitações para OPT

$\mathrm{OPT} =$ menor tempo de conclusão de um escalonamento

- Duração da tarefa mais longa:

$$\mathrm{OPT} \geq \max\{d[1], d[2], \ldots, d[n]\}$$

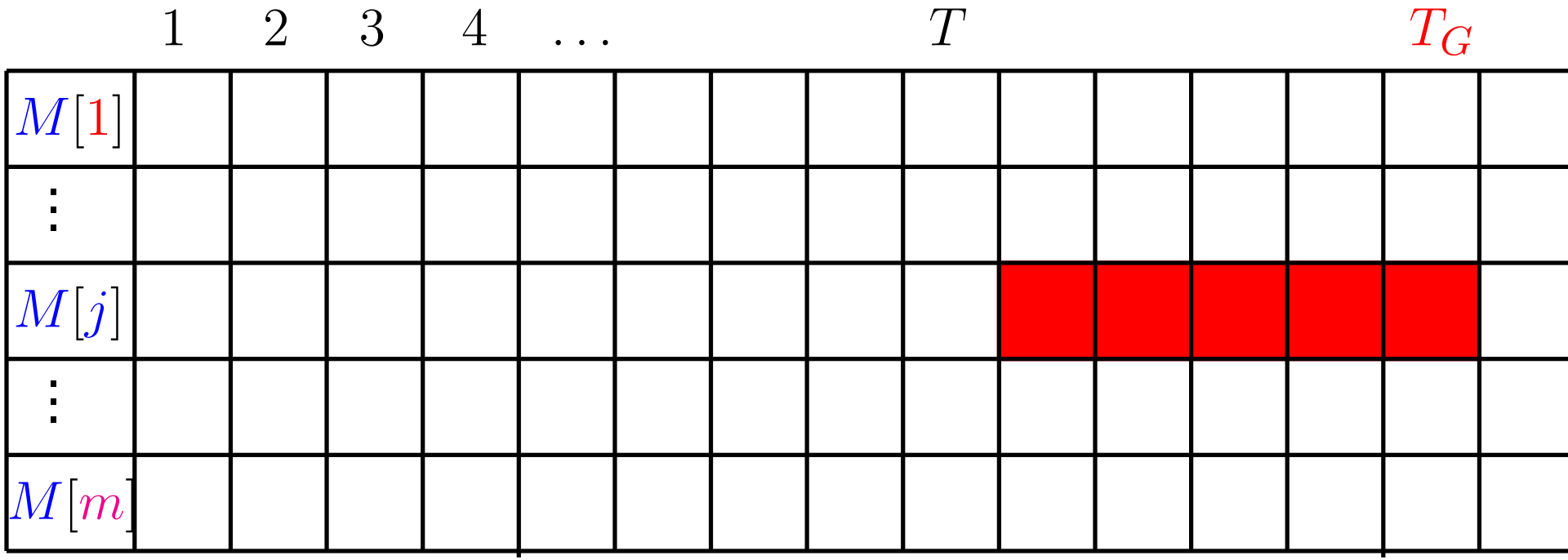
- Distribuição balanceada:

$$\mathrm{OPT} \geq \frac{d[1] + d[2] + \cdots + d[n]}{m}$$

Fator de aproximação

tarefa n é executada na máquina j a partir do instante T

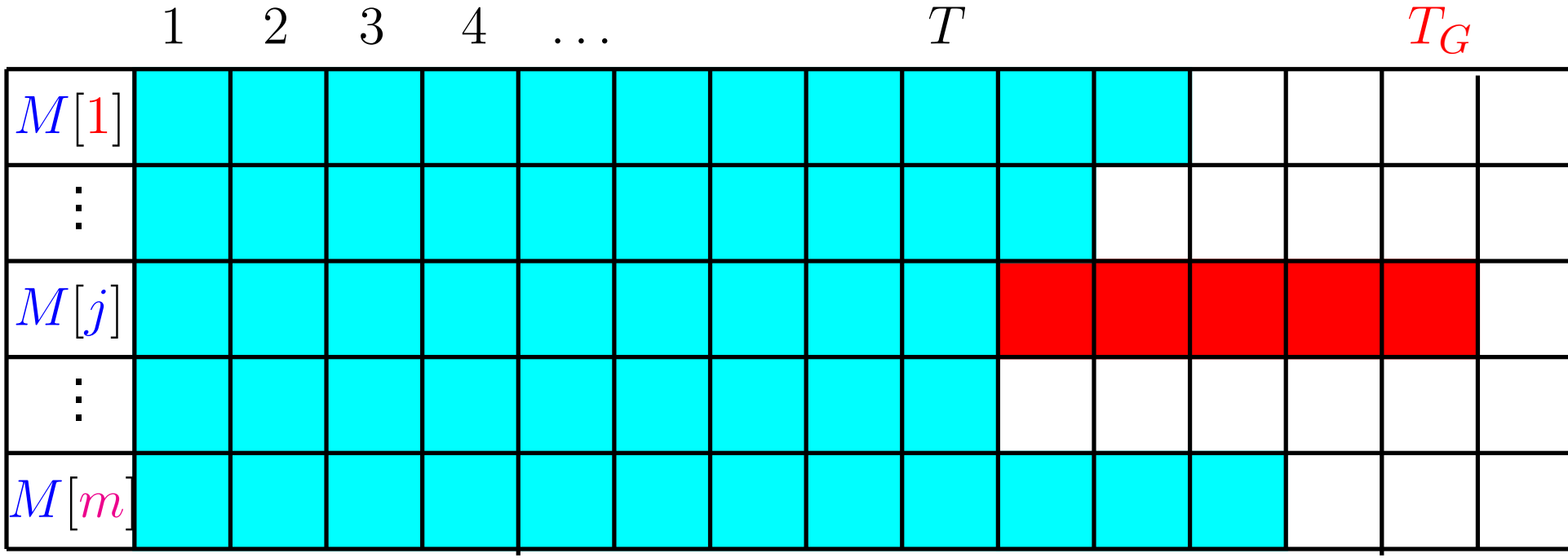
$T_G =$ conclusão do escalonamento obtido pelo algoritmo



Fator de aproximação (cont.)

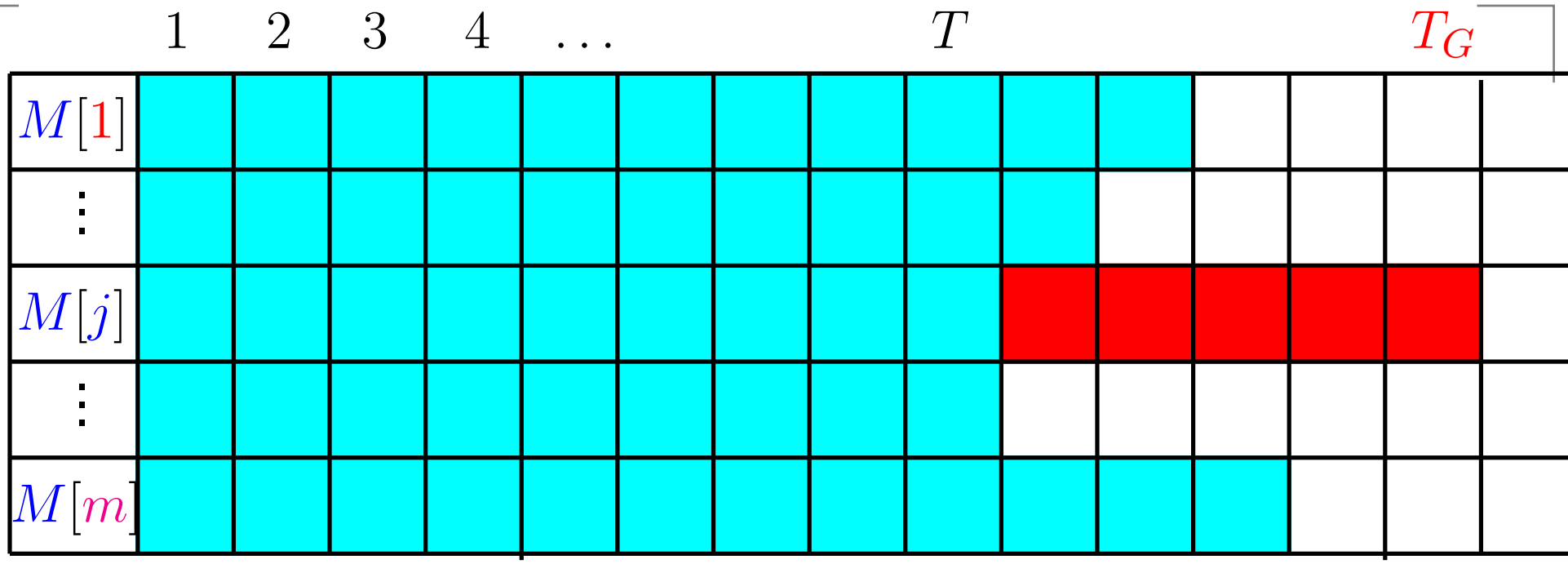
tarefa n é executada na máquina j a partir do instante T

$T_G =$ conclusão do escalonamento obtido pelo algoritmo



$$T \cdot m \leq d[1] + \cdots + d[n] \quad \Rightarrow \quad T \leq \frac{d[1] + \cdots + d[n]}{m}$$

Fator de aproximação (cont.)



$$
\begin{aligned}
T_G &= T + d[n] \\
&\leq \frac{d[1] + \cdots + d[n]}{m} + d[n] \\
&\leq \frac{d[1] + \cdots + d[n]}{m} + \max\{d[1], \ldots, d[n]\} \\
&\leq \mathrm{OPT} + \mathrm{OPT} = 2\,\mathrm{OPT}
\end{aligned}
$$

Conclusão

ESCALONAMENTO-GRAHAM (m, n, d)

1 **para** $j \leftarrow 1$ **até** m **faça**
2 $\quad M[j] \leftarrow \emptyset$
3 $\quad T[j] \leftarrow 0$
4 **para** $i \leftarrow 1$ **até** n **faça**
5 $\quad$ seja k tal que $T[k]$ é mínimo
6 $\quad M[k] \leftarrow M[k] \cup \{i\}$
7 $\quad T[k] \leftarrow T[k] + d[i]$
8 **devolva** $\{M[1], \ldots, M[m]\}$

Teorema: O algoritmo ESCALONAMENTO-GRAHAM é uma 2-aproximação.

Uma melhora no algoritmo

Regra LPT: longest processing time rule

Ordene as tarefas pelo tempo de processamento, com as mais longas primeiro, e aplique ESCALONAMENTO-GRAHAM.

Uma melhora no algoritmo

Regra LPT: longest processing time rule

Ordene as tarefas pelo tempo de processamento,
com as mais longas primeiro, e aplique
ESCALONAMENTO-GRAHAM.

Mostre que o algoritmo resultante é uma $4/3$-aproximação.

Uma melhora no algoritmo

Regra LPT: longest processing time rule

Ordene as tarefas pelo tempo de processamento,
com as mais longas primeiro, e aplique
ESCALONAMENTO-GRAHAM.

Mostre que o algoritmo resultante é uma $4/3$-aproximação.

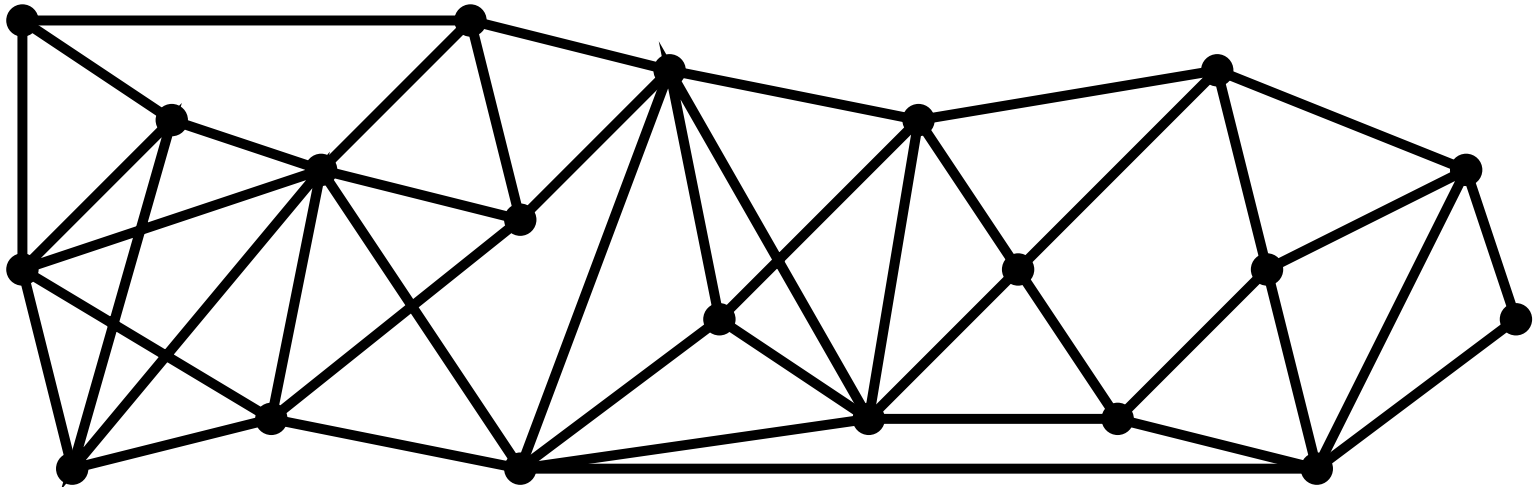
Mais para frente vamos ver um PTAS para este problema.

Problema do Caixeiro Viajante

Dados

grafo G

comprimento l_{ij} da aresta ij $\quad (ij \in E_G)$

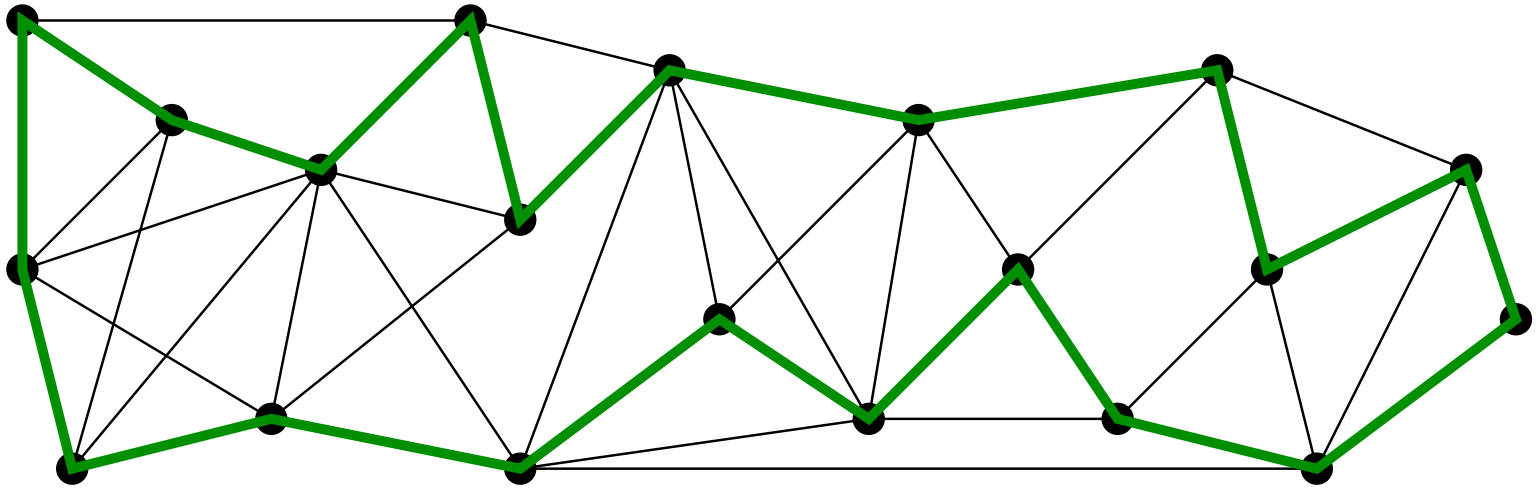


Problema do Caixeiro Viajante

Dados

grafo G

comprimento l_{ij} da aresta ij $\quad (ij \in E_G)$



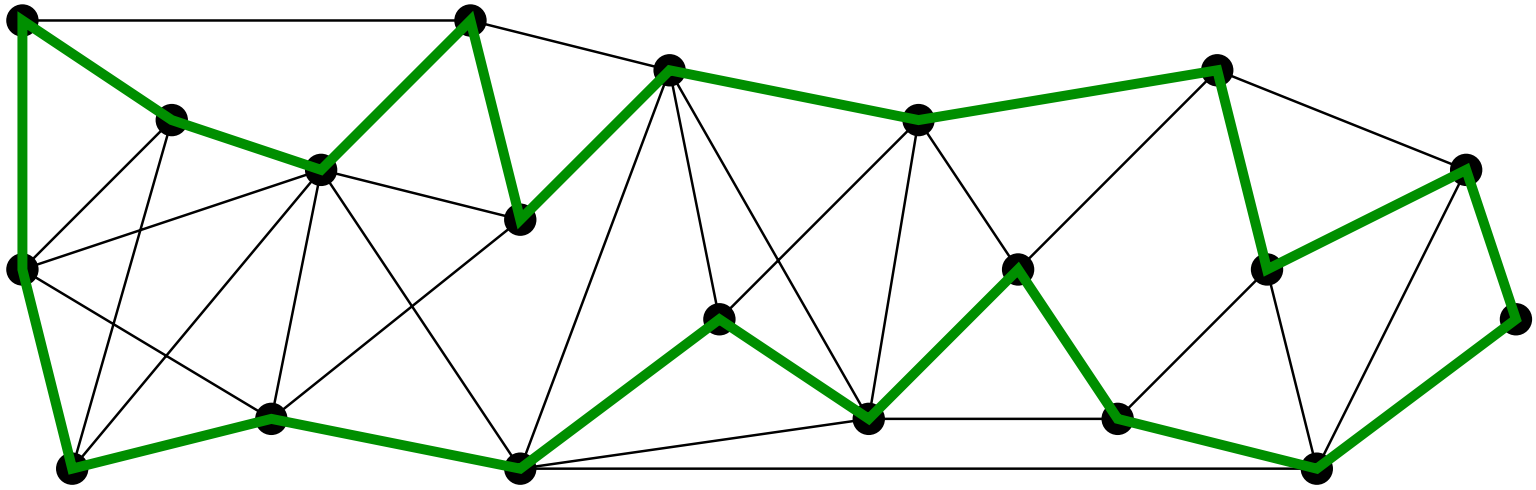
Circuito hamiltoniano: circuito que passa por todos os vértices

Problema do Caixeiro Viajante

Dados

grafo G

comprimento l_{ij} da aresta ij $\quad (ij \in E_G)$

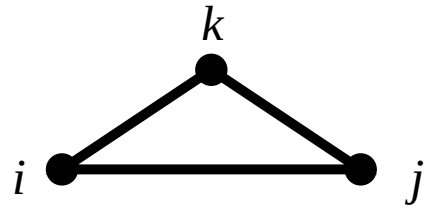


Circuito hamiltoniano: circuito que passa por todos os vértices

Problema (TSP): Dados G e l, encontrar circuito hamiltoniano C em G de comprimento $l(C)$ mínimo.

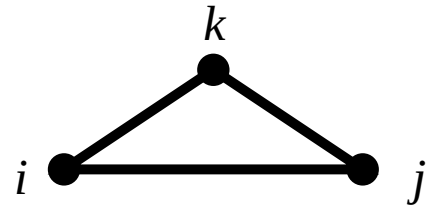
Variantes do Caixeiro Viajante

- TSP métrico
  - grafo completo
  - função comprimento l satisfaz desigualdade triangular: $l_{ij} \leq l_{ik} + l_{kj} \quad \forall i, j, k \in V_G$



Variantes do Caixeiro Viajante

- TSP métrico
  - grafo completo
  - função comprimento l satisfaz desigualdade triangular: $l_{ij} \leq l_{ik} + l_{kj} \quad \forall i, j, k \in V_G$



- TSP euclideano
  - Caso particular do métrico
  - Vértices são pontos no plano
    (ou num espaço euclideano qq de dimensão fixa)
  - l_{ij} é a distância euclideana entre i e j

Resultados Conhecidos

- NP-difícil mesmo se $l_e \in \{1, 2\}$ para toda aresta e [GJ79]
- Difícil de aproximar [SG76]
- $3/2$-aproximação para o caso métrico [C76]
- PTAS para o caso euclideano [A98,M99]

Resultados Conhecidos

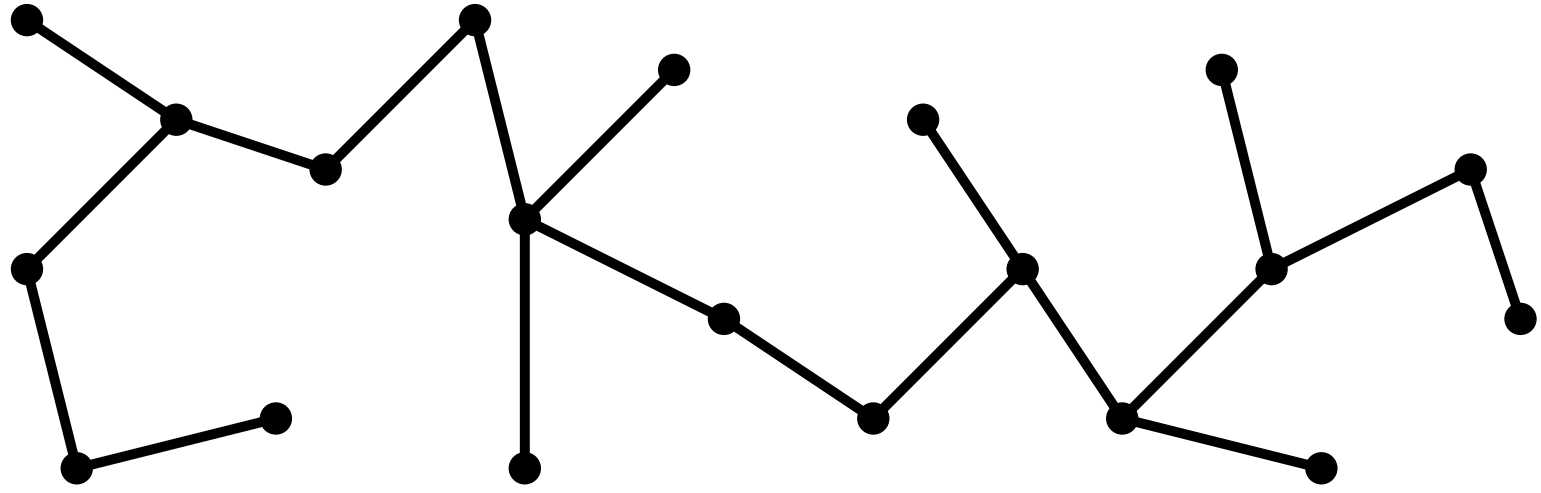
- NP-difícil mesmo se $l_e \in \{1, 2\}$ para toda aresta e [GJ79]
- Difícil de aproximar [SG76]
- $3/2$-aproximação para o caso métrico [C76]
- PTAS para o caso euclideano [A98,M99]

Nesta aula:

- 2-aproximação para o caso métrico [RSL77]
- $3/2$-aproximação para o caso métrico
- Comentários sobre o PTAS do caso euclideano
- TSP é difícil de aproximar

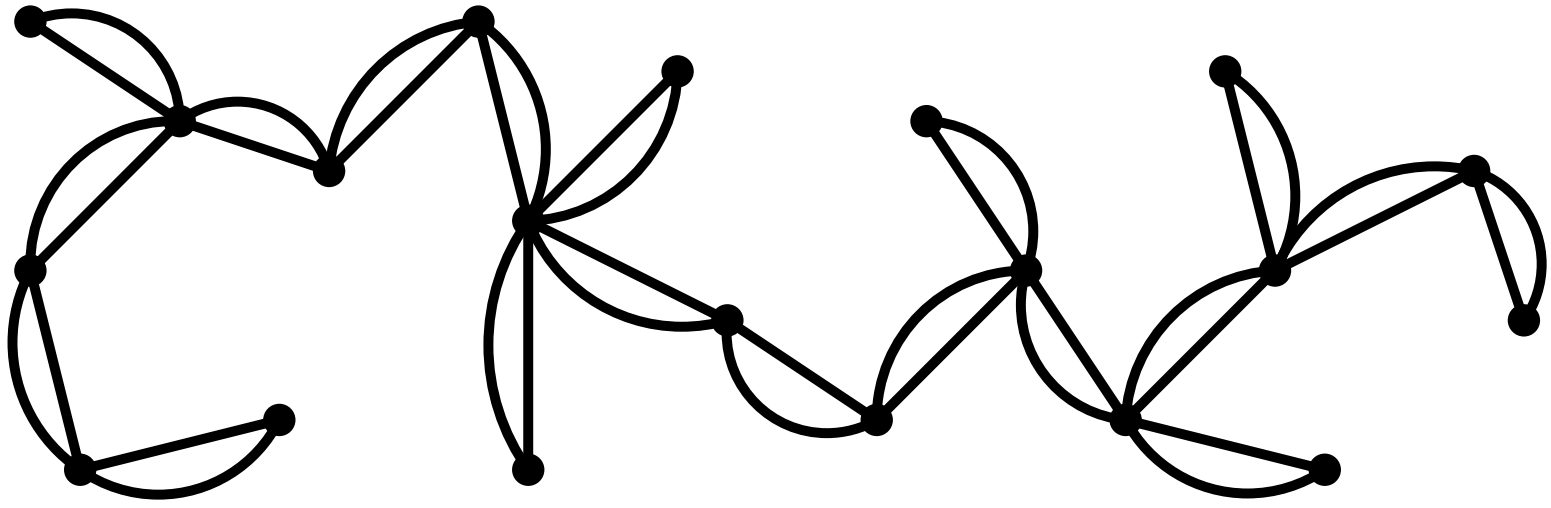
2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)



2-aproximação p/o TSP Métrico

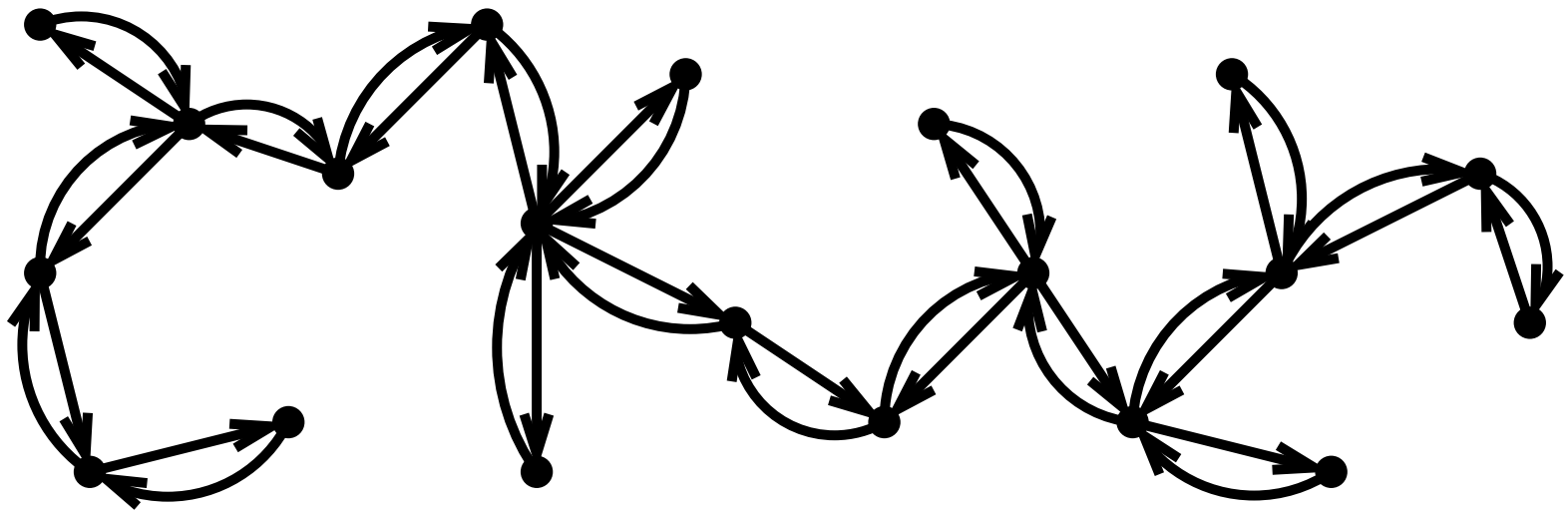
(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)



2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

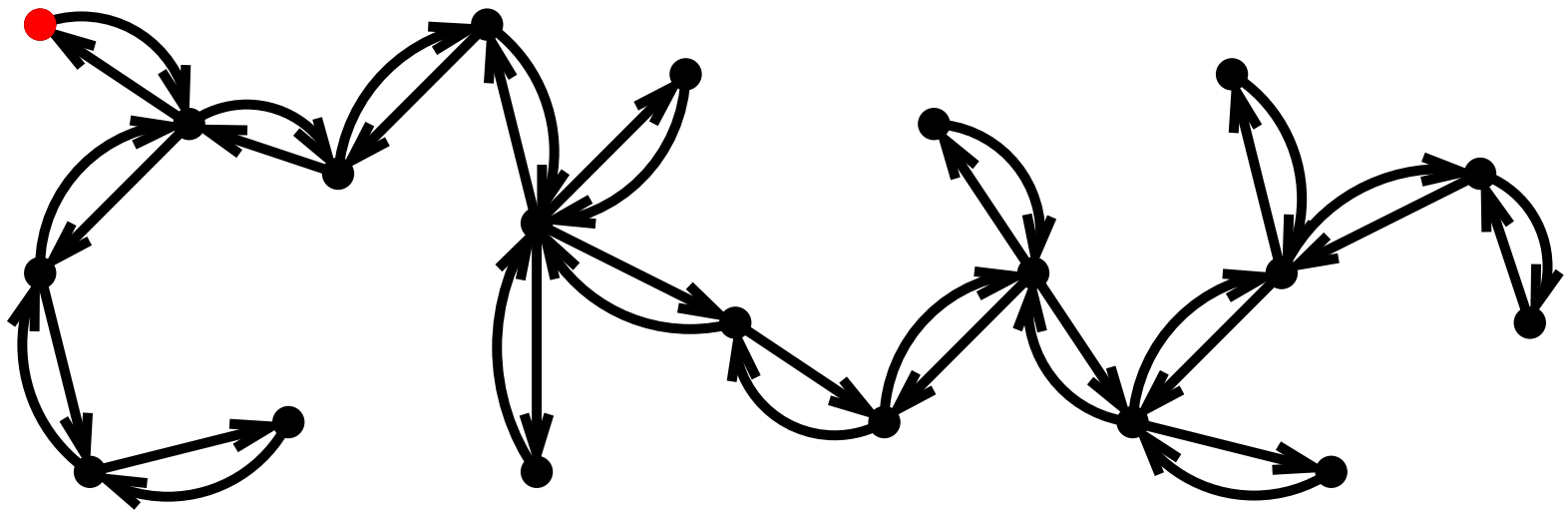


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito hamiltoniano C

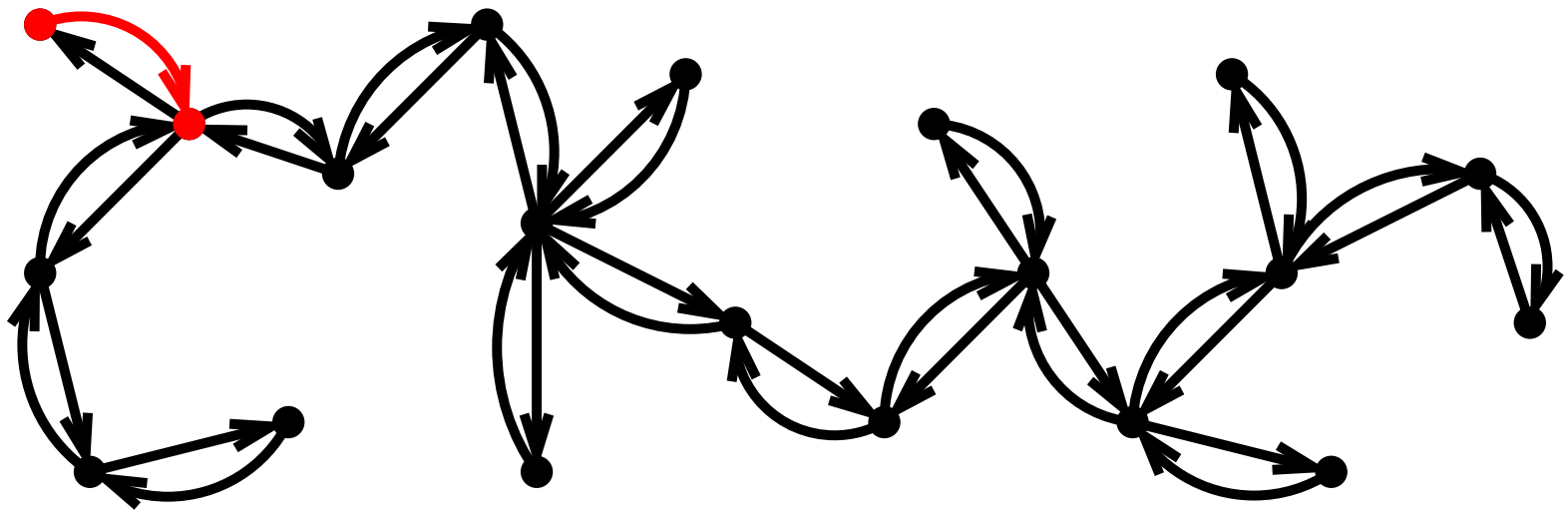


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito hamiltoniano C

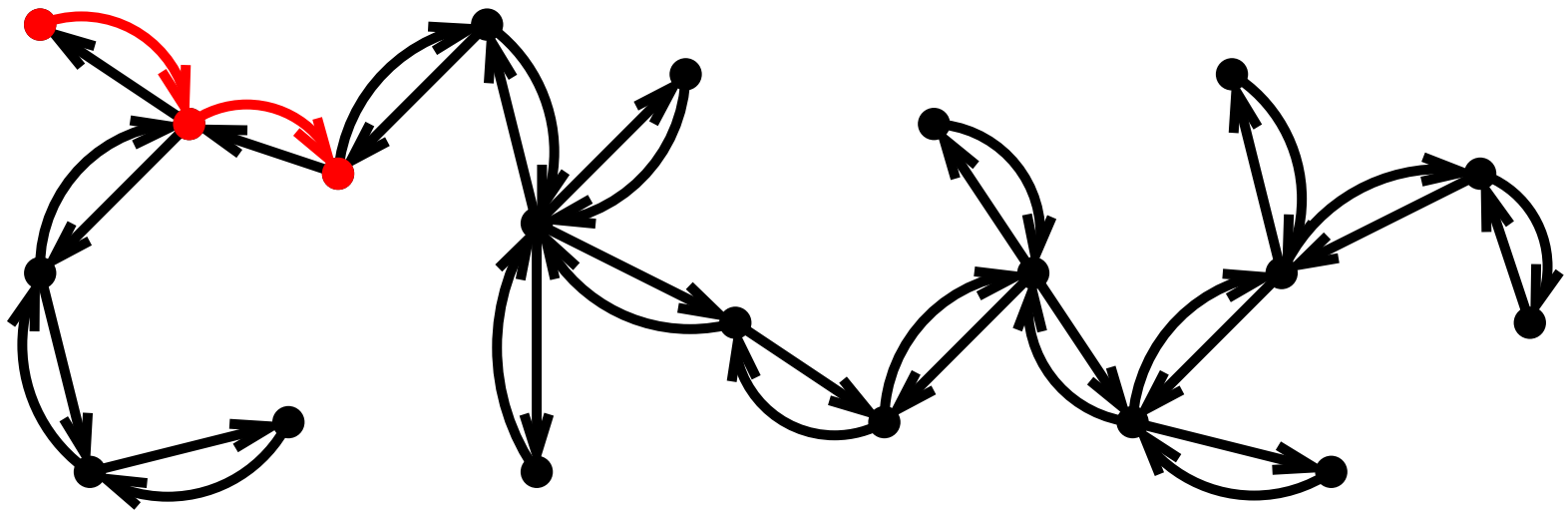


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito hamiltoniano C

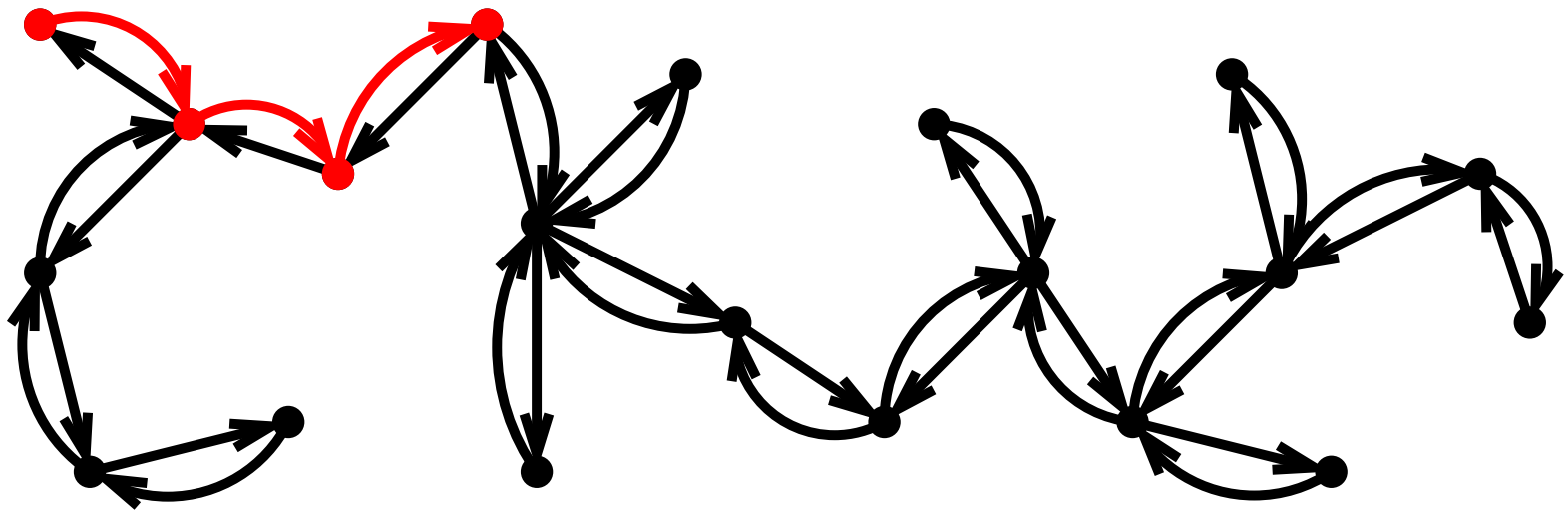


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito hamiltoniano C

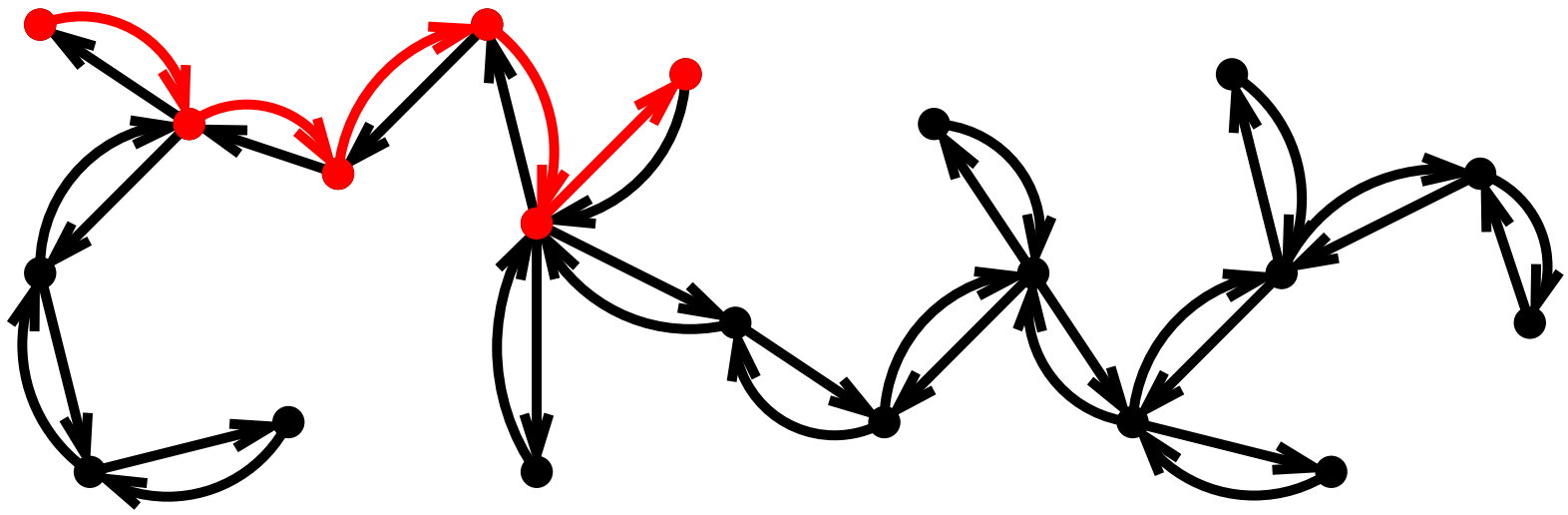


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito hamiltoniano C

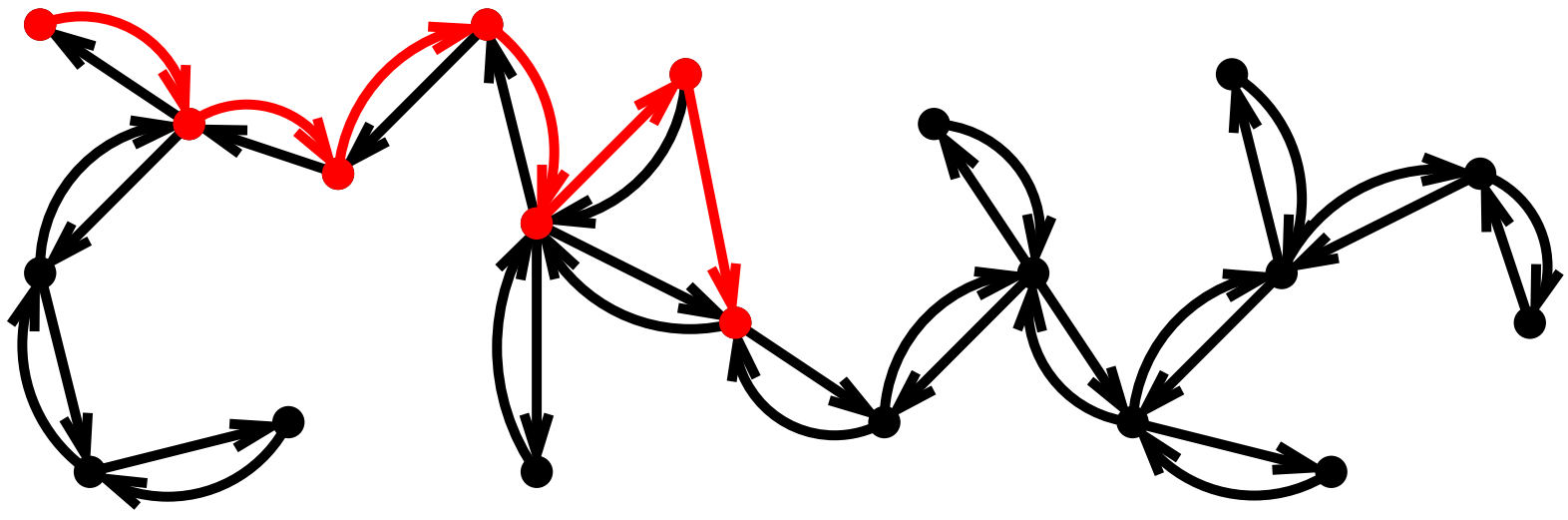


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito hamiltoniano C

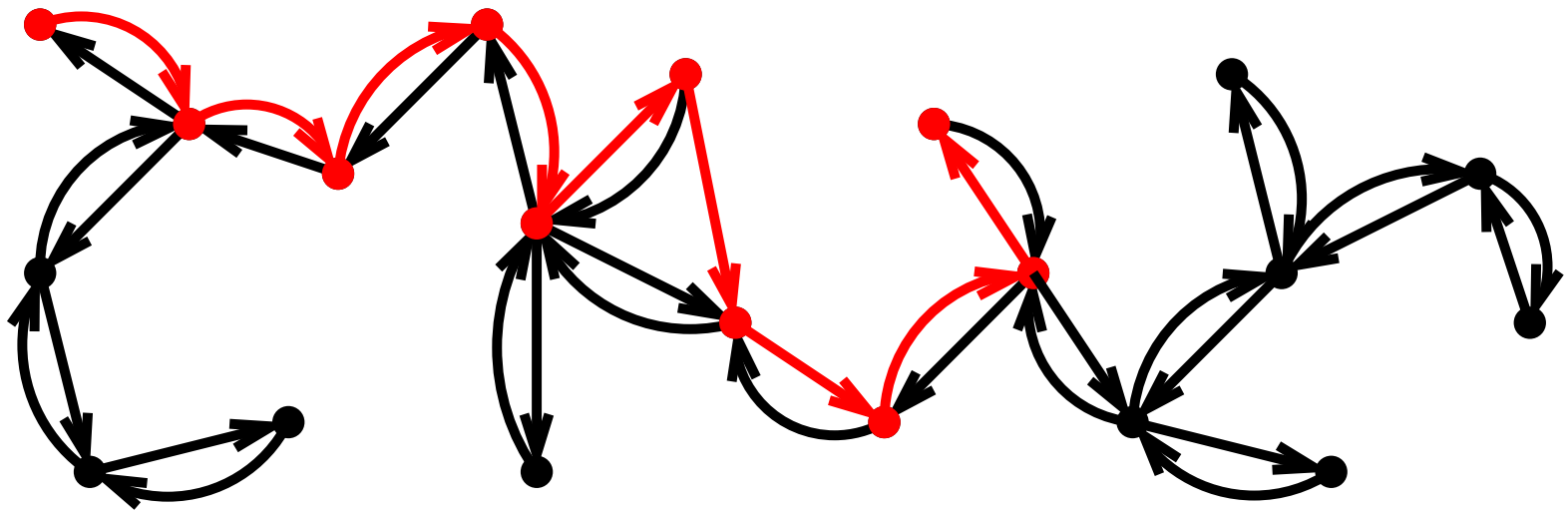


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito hamiltoniano C

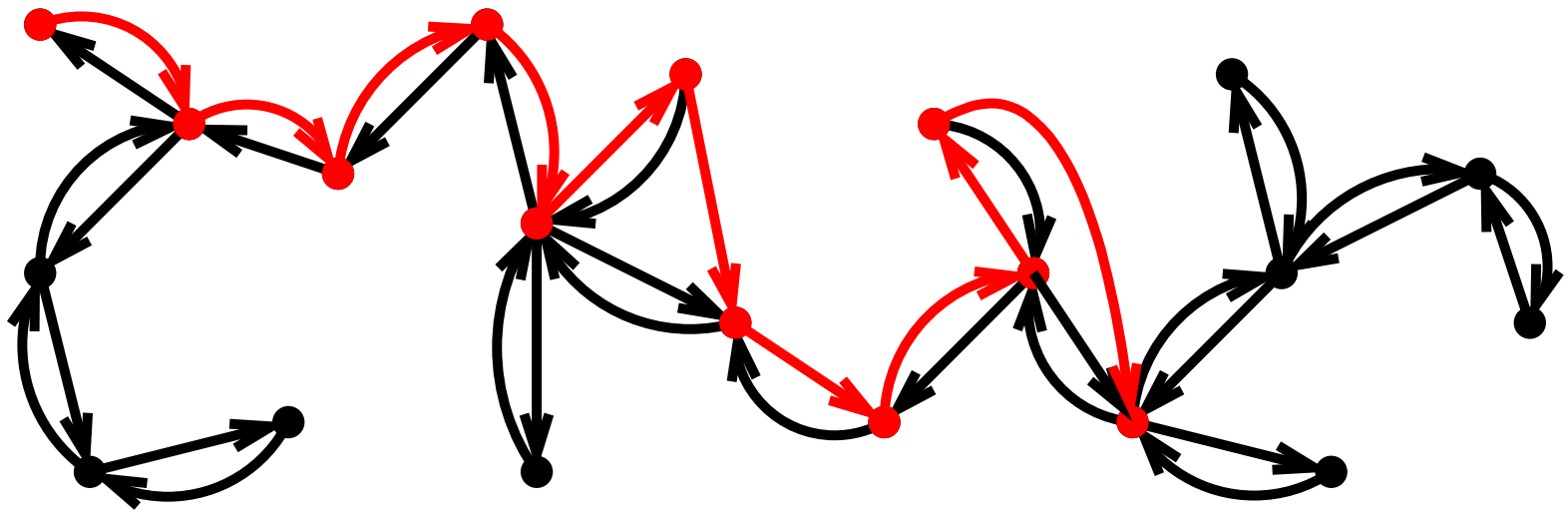


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito hamiltoniano C

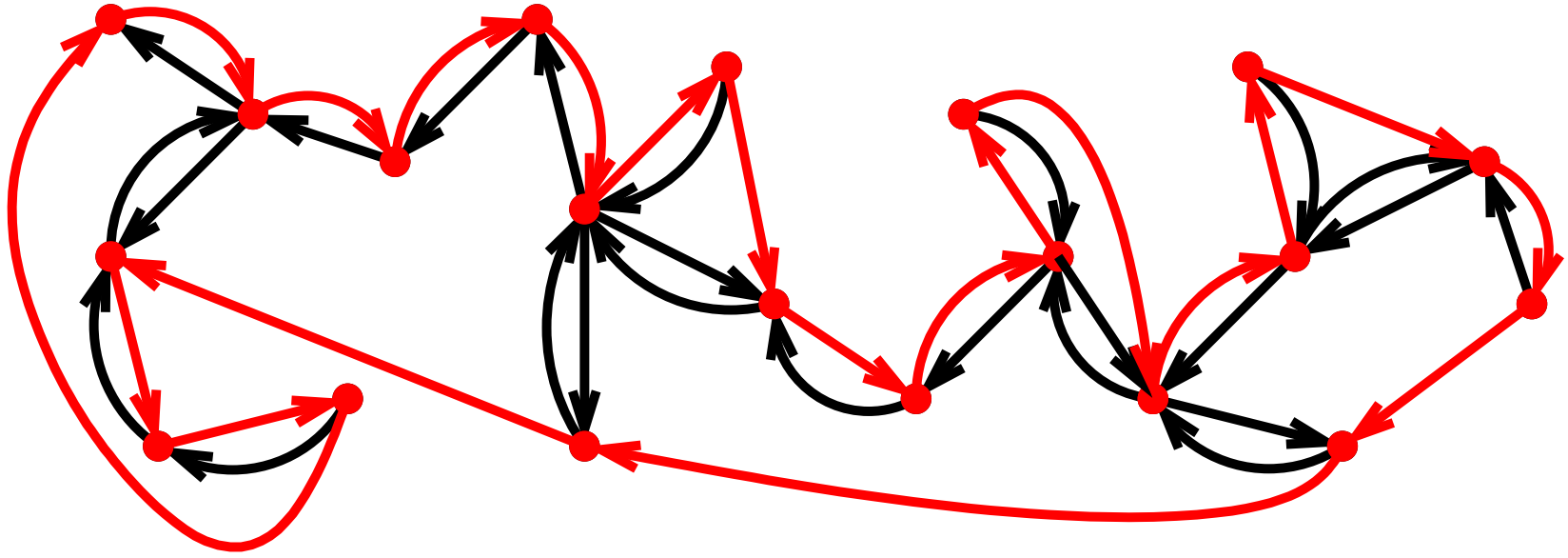


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito hamiltoniano C

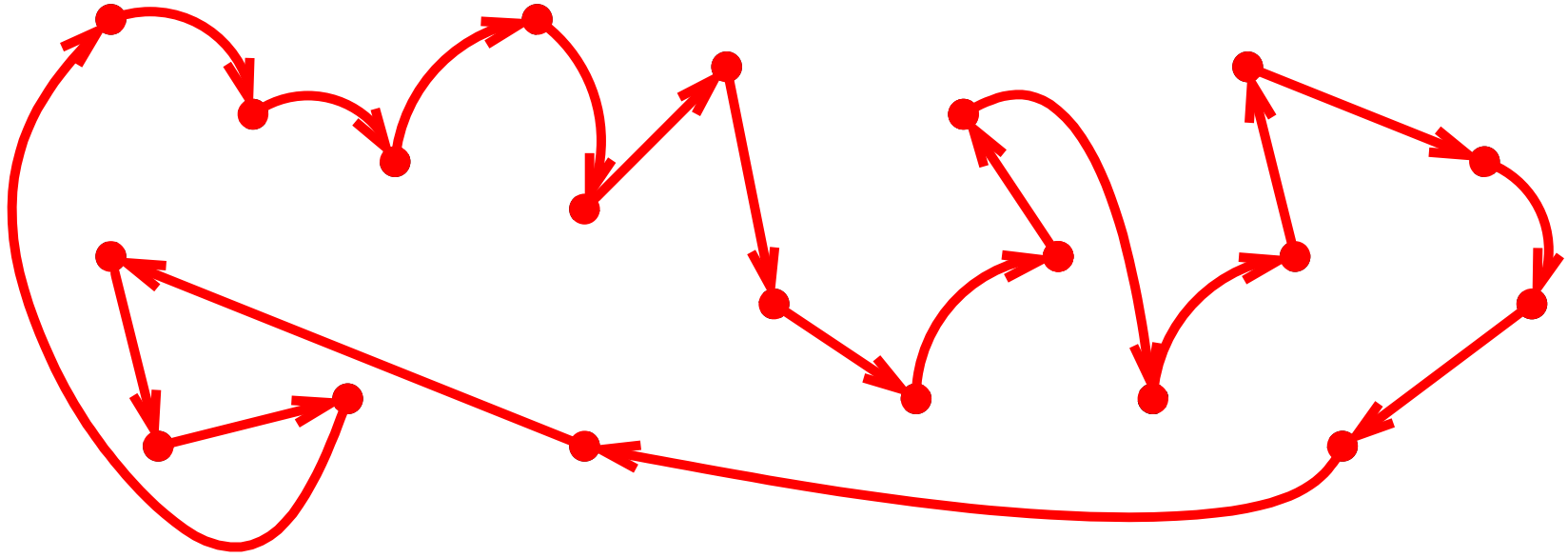


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo euleriano P (polinomial)

ciclo euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito hamiltoniano C
(5) Devolva C



2-Aproximação p/o TSP Métrico

Algoritmo TSPM-Rosenkrantz-Stearn-Lewis (G, l)

$T \leftarrow \textsf{MST}(G, l)$

$T' \leftarrow T + T$

$P \leftarrow \textsf{EULER}(T')$

$C \leftarrow \textsf{ATALHO}(P)$

devolve C

2-Aproximação p/o TSP Métrico

Algoritmo TSPM-Rosenkrantz-Stearn-Lewis (G, l)

$T \leftarrow \textsf{MST}(G, l)$

$T' \leftarrow T + T$

$P \leftarrow \textsf{EULER}(T')$

$C \leftarrow \textsf{ATALHO}(P)$

devolve C

Tempo de execução: $m \log n$

n: o número de vértices de G
m: o número de arestas de G

2-Aproximação p/o TSP Métrico

Delimitação inferior para OPT: $\mathrm{OPT} \geq l(T)$

onde T é árvore geradora de comprimento mínimo

2-Aproximação p/o TSP Métrico

Delimitação inferior para OPT: $\mathrm{OPT} \geq l(T)$

onde T é árvore geradora de comprimento mínimo

Prova:

C^*: circuito hamiltoniano de comprimento mínimo

e: aresta de C^*

$C^* - e$ é árvore geradora

2-Aproximação p/o TSP Métrico

Delimitação inferior para OPT: $\mathrm{OPT} \geq l(T)$

onde T é árvore geradora de comprimento mínimo

Prova:

C^*: circuito hamiltoniano de comprimento mínimo

e: aresta de C^*

$C^* - e$ é árvore geradora

$$\mathrm{OPT} = l(C^*) \geq l(C^* - e) \geq l(T). \quad \square$$

2-Aproximação p/o TSP Métrico

Delimitação inferior para OPT: $\mathrm{OPT} \geq l(T)$

onde T é árvore geradora de comprimento mínimo

Prova:

C^*: circuito hamiltoniano de comprimento mínimo

e: aresta de C^*

$C^* - e$ é árvore geradora

$$\mathrm{OPT} = l(C^*) \geq l(C^* - e) \geq l(T). \quad \square$$

Teorema: TSPM-Rosenkrantz-Stearn-Lewis é uma 2-aproximação polinomial para o TSP métrico.

2-Aproximação p/o TSP Métrico

Delimitação inferior para OPT: $\mathrm{OPT} \geq l(T)$

onde T é árvore geradora de comprimento mínimo

Prova:

C^*: circuito hamiltoniano de comprimento mínimo

e: aresta de C^*

$C^* - e$ é árvore geradora

$$\mathrm{OPT} = l(C^*) \geq l(C^* - e) \geq l(T). \quad \square$$

Teorema: TSPM-Rosenkrantz-Stearn-Lewis é uma 2-aproximação polinomial para o TSP métrico.

Prova: $l(C) \leq l(P) = l(T') = 2\,l(T) \leq 2\,\mathrm{OPT}. \quad \square$