Dois problemas sobre carregamento de veículos eléctricos

mobiOS

ENSPM 2014 - 15 de Julho

Este trabalho foi apoiado pelo Projeto mobiOS do QREN/AdI

COMPETE
PROGRAMA OPERACIONAL FACTORES DE COMPETITIVIDADE

QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL
PORTUGAL 2007.2013

UNIÃO EUROPEIA
Fundo Europeu de Desenvolvimento Regional

O Projecto

O que é o MobiOS?

MobiOS é uma plataforma de software em desenvolvimento pela Critical Software para aplicações de mobilidade, em especial mobilidade eléctrica.

O Projecto

O que é o MobiOS?

MobiOS é uma plataforma de software em desenvolvimento pela Critical Software para aplicações de mobilidade, em especial mobilidade eléctrica.

Qual o nosso papel no projecto?

Foi-nos pedido para desenvolvermos soluções para dois problemas concretos:

- Planear uma rede de postos de carregamento;
- calendarizar localmente os carregamentos.

Planeamento de uma rede de carregamento de veículos eléctricos

Joana Cavadas*, Gonçalo Correia+, João Gouveia*

*DMUC - Universidade de Coimbra, + DEC - Universidade de Coimbra/ TU Delft

Objectivo

- Encontrar a melhor localização para as estações de carregamento de forma a maximizar o nível de satisfação, sob um orçamento fixo.

Objectivo

- Encontrar a melhor localização para as estações de carregamento de forma a maximizar o nível de satisfação, sob um orçamento fixo.

Duas fases

- Modelar a procura;
- Distribuir os postos.

Objectivo

- Encontrar a melhor localização para as estações de carregamento de forma a maximizar o nível de satisfação, sob um orçamento fixo.

Duas fases

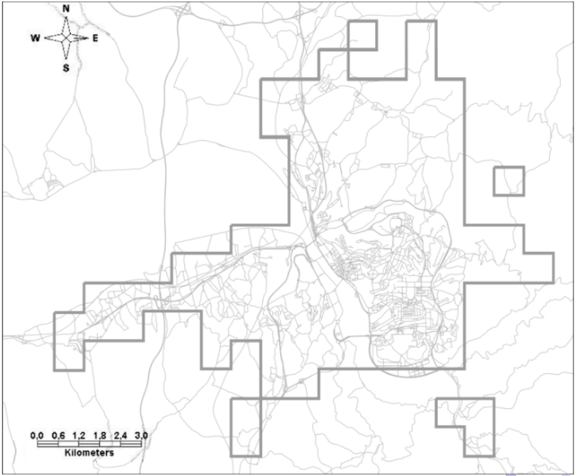
- Modelar a procura;
- Distribuir os postos.

Fase 1 : Modelar a procura

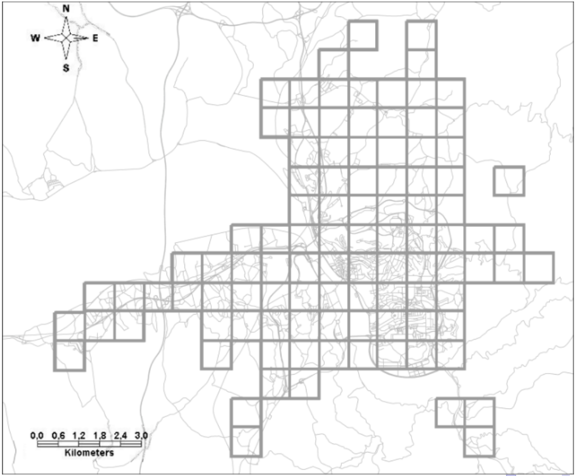
Baseado no tempo de estacionamento com duas simplificações:

- A maioria dos carros carrega (essentialmente) o mesmo número de vezes por dia.
- A probabilidade de um carro carregar é proporcional ao tempo que permanecerá no local.

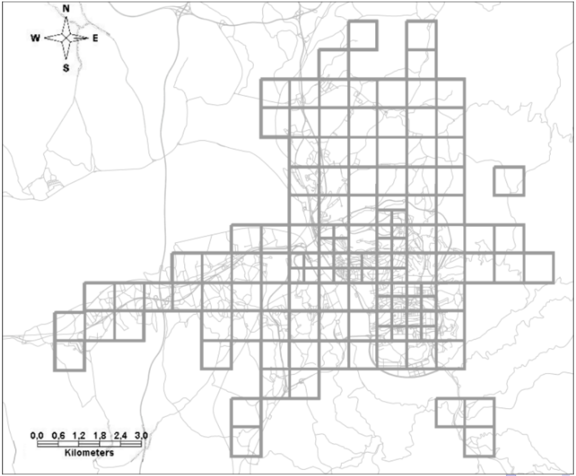
Modelação da procura - Exemplo



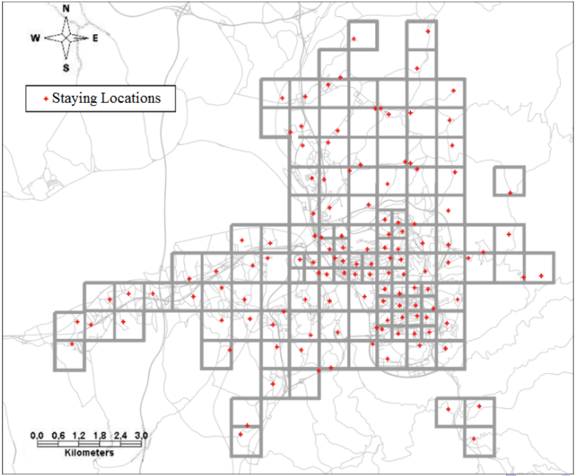
Modelação da procura - Exemplo



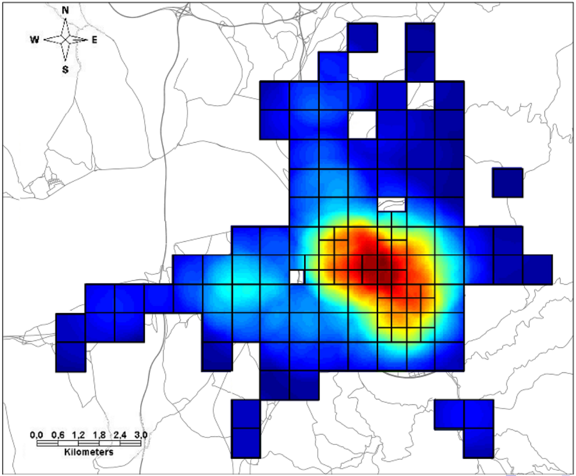
Modelação da procura - Exemplo



Modelação da procura - Exemplo



Modelação da procura - Exemplo



Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Estimámos em M pontos a procura D_j.

Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Estimámos em M pontos a procura D_j.

Criamos uma função Γ_{jk} que decresce à medida que a distância entre o ponto de procura j e o potencial posto k aumenta.

Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Estimámos em M pontos a procura D_j.

Criamos uma função Γ_{jk} que decresce à medida que a distância entre o ponto de procura j e o potencial posto k aumenta.

Obtemos:

Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Estimámos em M pontos a procura D_j.

Criamos uma função Γ_{jk} que decresce à medida que a distância entre o ponto de procura j e o potencial posto k aumenta.

Obtemos:

$x_k \in \{0, 1\},\ k = 1, \cdots, N$ ⋄ Construção ou não de k

Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Estimámos em M pontos a procura D_j.

Criamos uma função Γ_{jk} que decresce à medida que a distância entre o ponto de procura j e o potencial posto k aumenta.

Obtemos:

$z_{jk} \in [0,1], \; j = 1, \cdots M, \; k = 1, \cdots, N$ ◇ Procura do ponto j satisfeita em k

$x_k \in \{0,1\}, \; k = 1, \cdots, N$ ◇ Construção ou não de k

Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Estimámos em M pontos a procura D_j.

Criamos uma função Γ_{jk} que decresce à medida que a distância entre o ponto de procura j e o potencial posto k aumenta.

Obtemos:

max $\sum_{j=1}^{M} D_j \sum_{k=1}^{N} z_{jk}\Gamma_{jk}$ ⋄ Maximizar a procura satisfeita

t.q.

$z_{jk} \in [0,1], \; j = 1, \cdots M, \; k = 1, \cdots, N$ ⋄ Procura do ponto j satisfeita em k

$x_k \in \{0,1\}, \; k = 1, \cdots, N$ ⋄ Construção ou não de k

Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Estimámos em M pontos a procura D_j.

Criamos uma função Γ_{jk} que decresce à medida que a distância entre o ponto de procura j e o potencial posto k aumenta.

Obtemos:

max $\sum_{j=1}^{M} D_j \sum_{k=1}^{N} z_{jk}\Gamma_{jk}$ ◇ Maximizar a procura satisfeita

t.q.

$\sum_{k=1}^{n} x_k \leq T$ ◇ O orçamento não é excedido

$z_{jk} \in [0,1],\ j = 1, \cdots M,\ k = 1, \cdots, N$ ◇ Procura do ponto j satisfeita em k

$x_k \in \{0,1\},\ k = 1, \cdots, N$ ◇ Construção ou não de k

Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Estimámos em M pontos a procura D_j.

Criamos uma função Γ_{jk} que decresce à medida que a distância entre o ponto de procura j e o potencial posto k aumenta.

Obtemos:

$$\max \quad \sum_{j=1}^{M} D_j \sum_{k=1}^{N} z_{jk}\Gamma_{jk}$$ ◇ Maximizar a procura satisfeita

t.q.

$z_{jk} \leq x_k, \ j = 1, \cdots M, \ k = 1, \cdots, N$ ◇ Apenas postos construídos são usados

$\sum_{k=1}^{n} x_k \leq T$ ◇ O orçamento não é excedido

$z_{jk} \in [0, 1], \ j = 1, \cdots M, \ k = 1, \cdots, N$ ◇ Procura do ponto j satisfeita em k

$x_k \in \{0, 1\}, \ k = 1, \cdots, N$ ◇ Construção ou não de k

Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Estimámos em M pontos a procura D_j.

Criamos uma função Γ_{jk} que decresce à medida que a distância entre o ponto de procura j e o potencial posto k aumenta.

Obtemos:

max	$\sum_{j=1}^{M} D_j \sum_{k=1}^{N} z_{jk}\Gamma_{jk}$	◇ Maximizar a procura satisfeita
t.q.		
	$\sum_{k=1}^{N} D_i z_{ik} \leq D_i,\ \ i = 1, \cdots, M$	◇ Satisfaz apenas a procura estimada
	$z_{jk} \leq x_k,\ \ j = 1, \cdots M,\ \ k = 1, \cdots, N$	◇ Apenas postos construídos são usados
	$\sum_{k=1}^{n} x_k \leq T$	◇ O orçamento não é excedido
	$z_{jk} \in [0, 1],\ \ j = 1, \cdots M,\ \ k = 1, \cdots, N$	◇ Procura do ponto j satisfeita em k
	$x_k \in \{0, 1\},\ \ k = 1, \cdots, N$	◇ Construção ou não de k

Fase 2 - Modelo base de localização de postos

Queremos construir T postos de um total de N localizações possíveis, cada um com capacidade C.

Estimámos em M pontos a procura D_j.

Criamos uma função Γ_{jk} que decresce à medida que a distância entre o ponto de procura j e o potencial posto k aumenta.

Obtemos:

max	$\sum_{j=1}^{M} D_j \sum_{k=1}^{N} z_{jk}\Gamma_{jk}$	◇ Maximizar a procura satisfeita
t.q.	$\sum_{j=1}^{M} D_j z_{jk} \leq C, \;\; k = 1, \cdots, N$	◇ Capacidade não é excedida
	$\sum_{k=1}^{N} D_i z_{ik} \leq D_i, \;\; i = 1, \cdots, M$	◇ Satisfaz apenas a procura estimada
	$z_{jk} \leq x_k, \;\; j = 1, \cdots M, \;\; k = 1, \cdots, N$	◇ Apenas postos construídos são usados
	$\sum_{k=1}^{n} x_k \leq T$	◇ O orçamento não é excedido
	$z_{jk} \in [0, 1], \;\; j = 1, \cdots M, \;\; k = 1, \cdots, N$	◇ Procura do ponto j satisfeita em k
	$x_k \in \{0, 1\}, \;\; k = 1, \cdots, N$	◇ Construção ou não de k

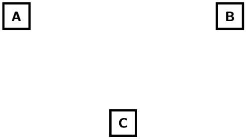
Alterações ao modelo base

Usaremos o modelo base apresentado com duas alterações essenciais:

Alterações ao modelo base

Usaremos o modelo base apresentado com duas alterações essenciais:

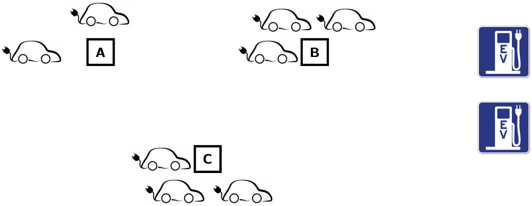
- Procura transferível;



Alterações ao modelo base

Usareмos o modelo base apresentado com duas alterações essenciais:

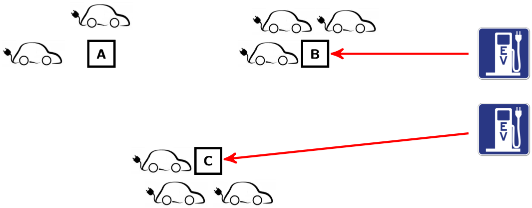
- Procura transferível;



Alterações ao modelo base

Usaremos o modelo base apresentado com duas alterações essenciais:

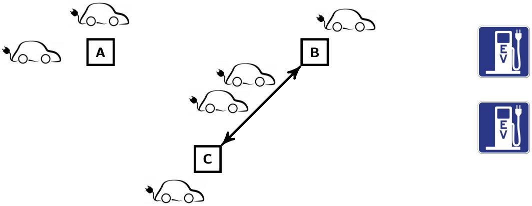
- Procura transferível;



Alterações ao modelo base

Usaremos o modelo base apresentado com duas alterações essenciais:

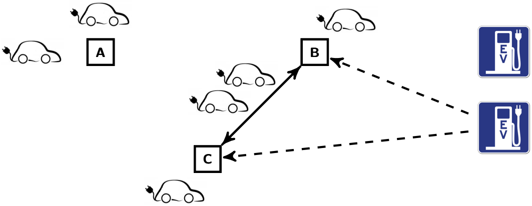
- Procura transferível;



Alterações ao modelo base

Usareмos o modelo base apresentado com duas alterações essenciais:

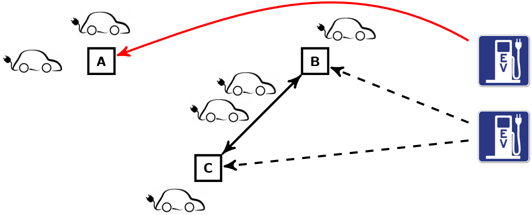
- Procura transferível;



Alterações ao modelo base

Usaremos o modelo base apresentado com duas alterações essenciais:

- Procura transferível;



Alterações ao modelo base

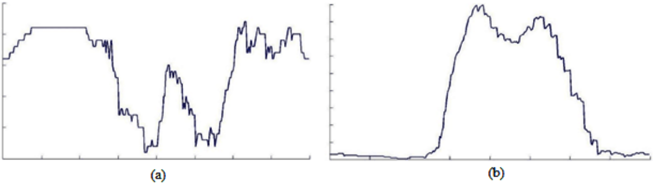
Usaremos o modelo base apresentado com duas alterações essenciais:

- Procura transferível;
- Divisão em intervalos de tempo.

Alterações ao modelo base

Usaremos o modelo base apresentado com duas alterações essenciais:

- Procura transferível;
- Divisão em intervalos de tempo.



Número de veículos estacionados numa área residecial (a)) e numa área comercial (b)) por hora do dia (Mondego Region Mobility System (2009))

Modelo com transferências (sem intervalos de tempo)

Modelo Básico **Modelo de transferência**

$$\max \quad \sum_{j=1}^{M} D_j \sum_{k=1}^{N} z_{jk} \cdot \Gamma_{jk} + \sum_{i=1}^{M} \sum_{j=1}^{M} V_{ij} \sum_{k=1}^{N} y_{ijk} \cdot \Gamma_{jk}$$

◇ Maximizar a procura satisfeita

$$\text{t.q.} \quad \sum_{j=1}^{M} D_j z_{jk} + \sum_{i=1}^{M} \sum_{j=1}^{M} V_{ij} y_{ijk} \leq C_k \quad , k=1,\ldots,N,$$

◇ Capacidade não é excedida

$$\sum_{k=1}^{N} D_i z_{ik} + \sum_{j=1}^{M} \sum_{k=1}^{N} W_{ij} y_{ijk} \leq D_i \quad , i=1,\ldots,M,$$

$$\sum_{k=1}^{N} y_{ijk} \leq 1, \quad i,j=1,\ldots,M,$$

◇ Satisfaz apenas a procura estimada

$$z_{jk} \leq x_k, \quad j=1,\ldots,M, \ k=1,\ldots,N,$$

$$y_{ijk} \leq x_k, \quad i,j=1,\ldots,M \ \ k=1,\ldots,N,$$

◇ Procura satisfeita por k $\Rightarrow$ Construção de k

$$\sum_{k=1}^{N} B_k x_k \leq T,$$

◇ O orçamento não é excedido

$$z_{jk} \in [0,1], \quad j=1,\ldots,M, \ k=1,\ldots,N,$$

◇ Procura do ponto j satisfeita em k

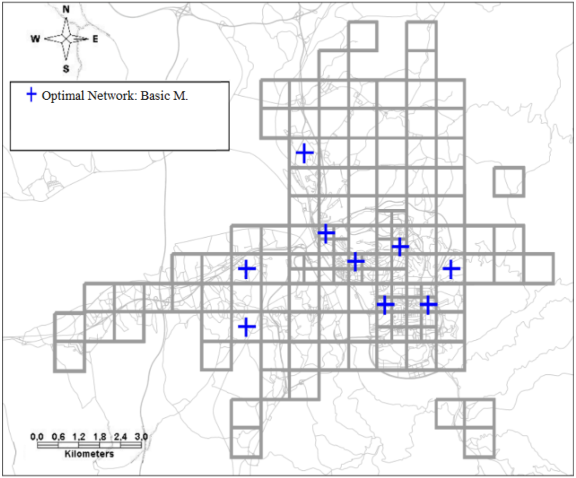
$$y_{ijk} \in [0,1], \quad i,j=1,\ldots,M, \quad k=1,\ldots,N,$$

◇ Procura transferida de i para j e satisfeita em k

$$x_k \in \{0,1\}, \quad k=1,\ldots,N.$$

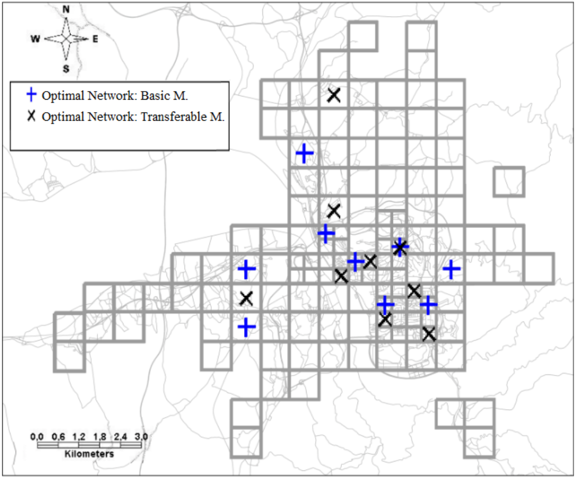
◇ Construção ou não de k

Redes optimais - Exemplo



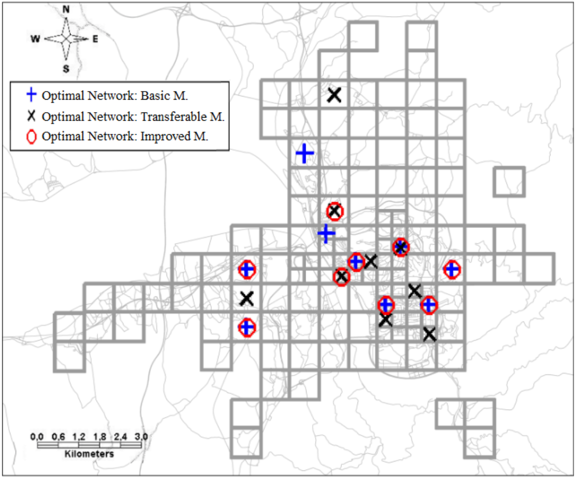
Solução do modelo base

Redes optimais - Exemplo



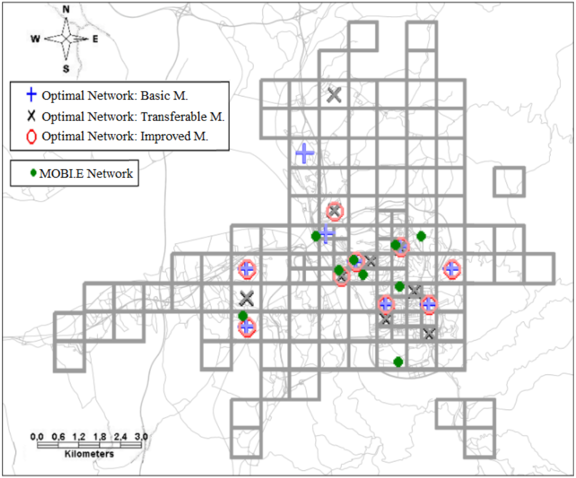
Solução com transferências

Redes optimais - Exemplo



Solução do modelo completo

Redes optimais - Exemplo



Rede Mobi.e

Optimização do escalonamento do carregamento de veículos eléctricos

Xavier Fernandes*, João Gouveia*, Rodrigo Maia+, Joana Rebelo*

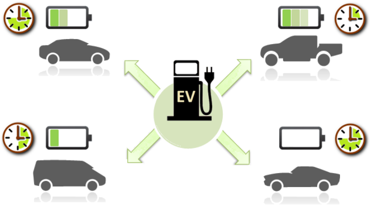
*DMUC - Universidade de Coimbra, + Critical Software

Objectivo:

- Dada uma potência fixa disponível, e veículos com necessidades individuais de carga e tempo disponível, desenhar um calendário optimizado de carregamento.

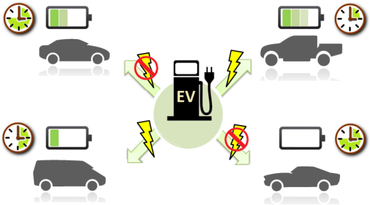
Objectivo:

- Dada uma potência fixa disponível, e veículos com necessidades individuais de carga e tempo disponível, desenhar um calendário optimizado de carregamento.



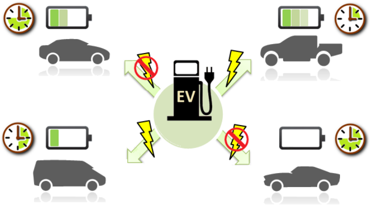
Objectivo:

- Dada uma potência fixa disponível, e veículos com necessidades individuais de carga e tempo disponível, desenhar um calendário optimizado de carregamento.



Objectivo:

- Dada uma potência fixa disponível, e veículos com necessidades individuais de carga e tempo disponível, desenhar um calendário optimizado de carregamento.



Notas:

- Cada vez que um carro chega temos de recalcular;
- queremos o melhor comportamento dinâmico possível.

Abordagem discreta: método das folgas

Uma primeira abordagem é dividir o tempo em intervalos. No início de cada intervalo verificamos a *folga* de cada carro (tempo disponível menos tempo necessário) e carregamos os carros com menor folga.

Abordagem discreta: método das folgas

Uma primeira abordagem é dividir o tempo em intervalos. No início de cada intervalo verificamos a *folga* de cada carro (tempo disponível menos tempo necessário) e carregamos os carros com menor folga.

Teorema

Se existir solução o método da folga devolve-a.

Abordagem discreta: método das folgas

Uma primeira abordagem é dividir o tempo em intervalos. No início de cada intervalo verificamos a *folga* de cada carro (tempo disponível menos tempo necessário) e carregamos os carros com menor folga.

Teorema

Se existir solução o método da folga devolve-a.

Esta é uma heurística de despacho usada com frequência em problemas de escalonamento, e que tem tipicamente boas propriedades dinâmicas.

Abordagem discreta: método das folgas

Uma primeira abordagem é dividir o tempo em intervalos. No início de cada intervalo verificamos a *folga* de cada carro (tempo disponível menos tempo necessário) e carregamos os carros com menor folga.

Teorema

Se existir solução o método da folga devolve-a.

Esta é uma heurística de despacho usada com frequência em problemas de escalonamento, e que tem tipicamente boas propriedades dinâmicas.

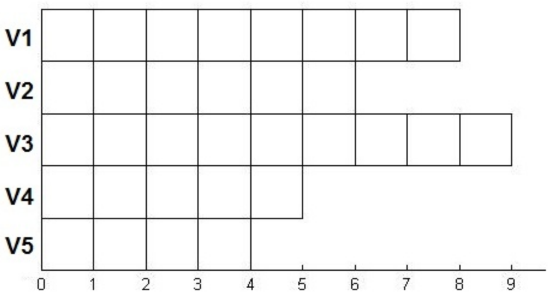
Vai servir de termo de comparação.

Exemplo

$T=(8,6,9,5,4)$
$C=(2,4,5,2,3)$
$P=3$

Exemplo

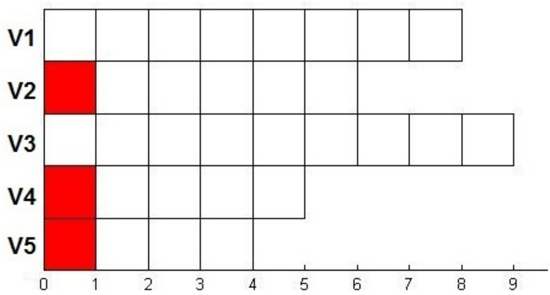
T=(8,6,9,5,4)
C=(2,4,5,2,3)
P=3



c=(2,4,5,2,3)
F=(6,2,4,3,1)

Exemplo

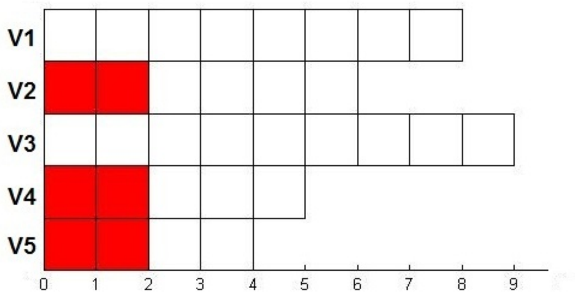
$T=(8,6,9,5,4)$
$C=(2,4,5,2,3)$
$P=3$



$c=(2,3,5,1,2)$
$F=(5,2,3,3,1)$

Exemplo

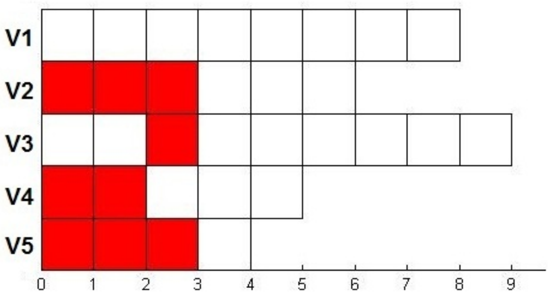
T=(8,6,9,5,4)
C=(2,4,5,2,3)
P=3



c=(2,2,5,0,1)
F=(4,2,2,3,1)

Exemplo

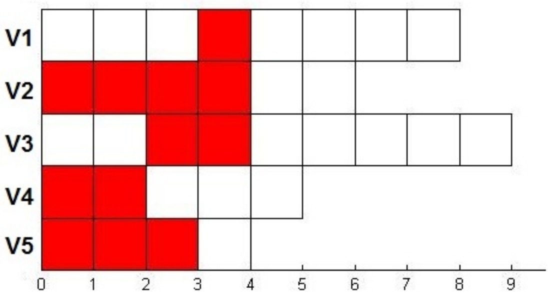
T=(8,6,9,5,4)
C=(2,4,5,2,3)
P=3



c=(2,1,4,0,0)
F=(3,2,2,3,1)

Exemplo

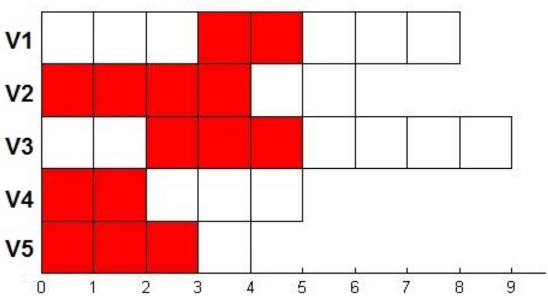
T=(8,6,9,5,4)
C=(2,4,5,2,3)
P=3



c=(1,0,3,0,0)
F=(3,2,2,3,1)

Exemplo

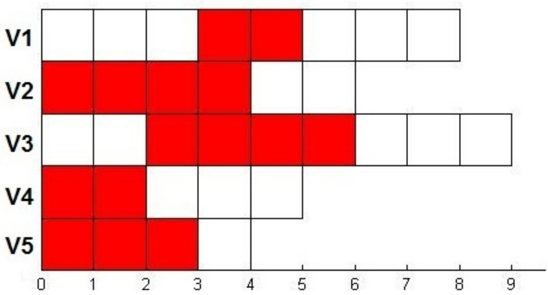
T=(8,6,9,5,4)
C=(2,4,5,2,3)
P=3



c=(0,0,2,0,0)
F=(3,2,2,3,1)

Exemplo

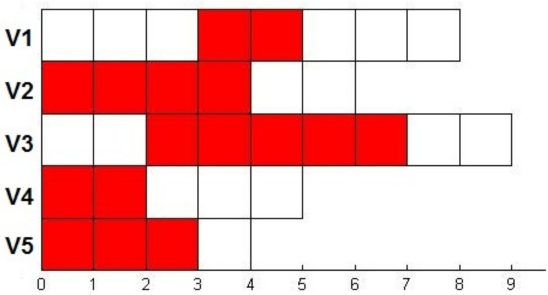
T=(8,6,9,5,4)
C=(2,4,5,2,3)
P=3



c=(0,0,1,0,0)
F=(3,2,2,3,1)

Exemplo

T=(8,6,9,5,4)
C=(2,4,5,2,3)
P=3



c=(0,0,0,0,0)
F=(3,2,2,3,1)

Abordagem contínua básica

Para K veículos e $i = 1, \cdots, K$ sejam

- $C_i \longrightarrow$ Tempo de carga necessário para o carro i
- $T_i \longrightarrow$ Tempo disponível para o carro i
- $P \longrightarrow$ Número máximo de carros que podem carregar simultaneamente.

Abordagem contínua básica

Para K veículos e $i = 1, \cdots, K$ sejam

- $C_i \longrightarrow$ Tempo de carga necessário para o carro i
- $T_i \longrightarrow$ Tempo disponível para o carro i
- $P \longrightarrow$ Número máximo de carros que podem carregar simultaneamente.

$y_{ij} \in [0,1] \longrightarrow$ fracção de $[T_{j-1}, T_j]$ em que i carrega

Abordagem contínua básica

Para K veículos e $i = 1, \cdots, K$ sejam

- $C_i \longrightarrow$ Tempo de carga necessário para o carro i
- $T_i \longrightarrow$ Tempo disponível para o carro i
- $P \longrightarrow$ Número máximo de carros que podem carregar simultaneamente.

$y_{ij} \in [0, 1] \longrightarrow$ fracção de $[T_{j-1}, T_j]$ em que i carrega

$$\begin{cases} \text{Encontrar } y_{ij} \text{ t.q.:} \\ \sum_{j=1}^{i} y_{ij}(T_j - T_{j-1}) = C_i, \; i = 1, ..., K; \\ \\ \end{cases}$$

Abordagem contínua básica

Para K veículos e $i = 1, \cdots, K$ sejam

- $C_i \longrightarrow$ Tempo de carga necessário para o carro i
- $T_i \longrightarrow$ Tempo disponível para o carro i
- $P \longrightarrow$ Número máximo de carros que podem carregar simultaneamente.

$y_{ij} \in [0, 1] \longrightarrow$ fracção de $[T_{j-1}, T_j]$ em que i carrega

$$\begin{cases} \text{Encontrar } y_{ij} \text{ t.q.:} \\ \sum_{j=1}^{i} y_{ij}(T_j - T_{j-1}) = C_i, \ i = 1, ..., K; \\ \sum_{i=j}^{K} y_{ij} \leq P, \ j = 1, ..., K; \end{cases}$$

Abordagem contínua básica

Para K veículos e $i = 1, \cdots, K$ sejam

- $C_i \longrightarrow$ Tempo de carga necessário para o carro i
- $T_i \longrightarrow$ Tempo disponível para o carro i
- $P \longrightarrow$ Número máximo de carros que podem carregar simultaneamente.

$y_{ij} \in [0,1] \longrightarrow$ fracção de $[T_{j-1}, T_j]$ em que i carrega

$$\begin{cases} \max \sum_{i=1}^{K} \sum_{j=1}^{i} \frac{1}{2^i} y_{ij}(T_j - T_{j-1}) \text{ t.q.:} \\ \sum_{j=1}^{i} y_{ij}(T_j - T_{j-1}) = C_i, \; i = 1, ..., K; \\ \sum_{i=j}^{K} y_{ij} \leq P, \; j = 1, ..., K; \end{cases}$$

Abordagem contínua básica

Para K veículos e $i = 1, \cdots, K$ sejam

- $C_i \longrightarrow$ Tempo de carga necessário para o carro i
- $T_i \longrightarrow$ Tempo disponível para o carro i
- $P \longrightarrow$ Número máximo de carros que podem carregar simultaneamente.

$y_{ij} \in [0, 1] \longrightarrow$ fracção de $[T_{j-1}, T_j]$ em que i carrega

$$\begin{cases} \max \sum_{i=1}^{K} \sum_{j=1}^{i} \frac{1}{2^i} y_{ij}(T_j - T_{j-1}) \text{ t.q.:} \\ \sum_{j=1}^{i} y_{ij}(T_j - T_{j-1}) = C_i, \ i = 1, ..., K; \\ \sum_{i=j}^{K} y_{ij} \leq P, \ j = 1, ..., K; \end{cases}$$

É um PL inspirado pelo método discreto das folgas.

Exemplo

Consideremos:

6 VE's P=3

T=[17 18 22 22 24 25]

C=[13 8 19 8 4 16]

Exemplo

Consideremos:

6 VE's P=3

T=[17 18 22 22 24 25]

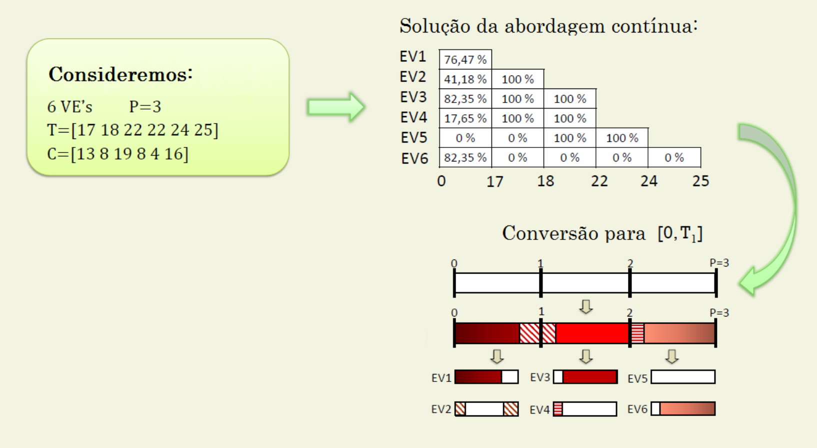
C=[13 8 19 8 4 16]

Solução da abordagem contínua:

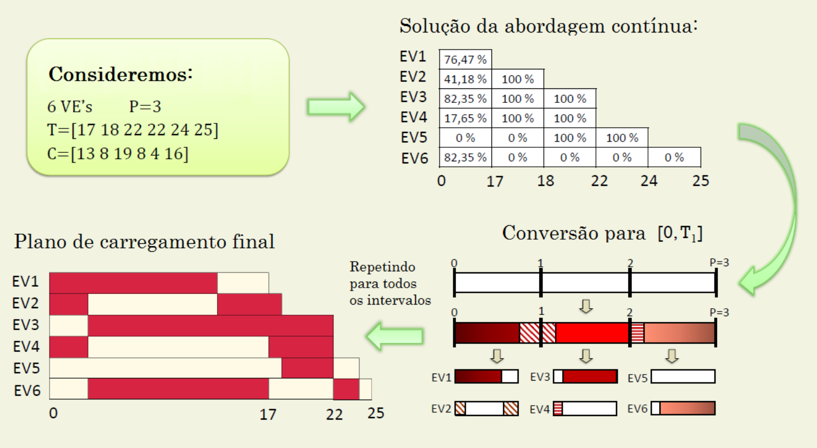
EV1	76,47 %				
EV2	41,18 %	100 %			
EV3	82,35 %	100 %	100 %		
EV4	17,65 %	100 %	100 %		
EV5	0 %	0 %	100 %	100 %	
EV6	82,35 %	0 %	0 %	0 %	0 %

0 17 18 22 24 25

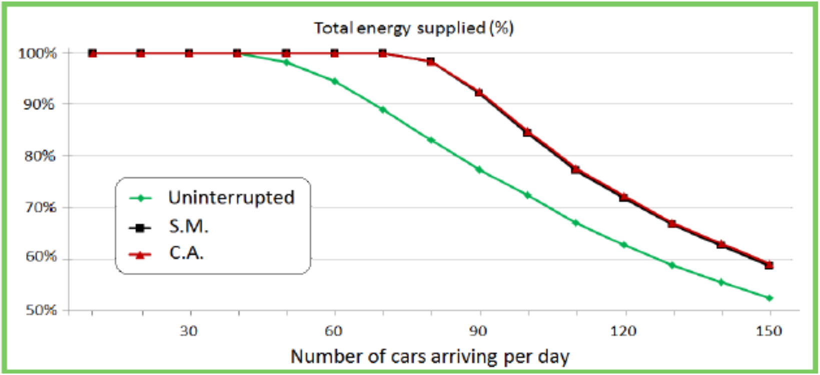
Exemplo


Consideremos:
6 VE's P=3
T=[17 18 22 22 24 25]
C=[13 8 19 8 4 16]
Solução da abordagem contínua:
EV1 76,47 %
EV2 41,18 % 100 %
EV3 82,35 % 100 % 100 %
EV4 17,65 % 100 % 100 %
EV5 0 % 0 % 100 % 100 %
EV6 82,35 % 0 % 0 % 0 % 0 %
0 17 18 22 24 25
Conversão para [0, T1]
0 1 2 P=3
0 1 2 P=3
EV1 EV3 EV5
EV2 EV4 EV6

Exemplo

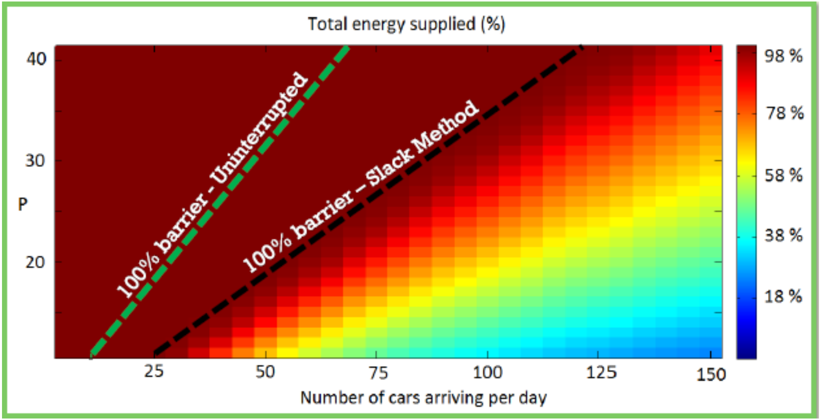

Consideremos:
6 VE's P=3
T=[17 18 22 22 24 25]
C=[13 8 19 8 4 16]
Solução da abordagem contínua:
EV1 76,47 %
EV2 41,18 % 100 %
EV3 82,35 % 100 % 100 %
EV4 17,65 % 100 % 100 %
EV5 0 % 0 % 100 % 100 %
EV6 82,35 % 0 % 0 % 0 % 0 %
0 17 18 22 24 25
Conversão para [0, T1]
0 1 2 P=3
Repetindo para todos os intervalos
EV1 EV3 EV5
EV2 EV4 EV6
Plano de carregamento final
EV1
EV2
EV3
EV4
EV5
EV6
0 17 22 25

Resultados da simulação



Comparação de métodos para $P = 25$ (1 ano simulado).

Resultados da simulação 2



Comparação dos métodos.

Obrigado