# Resolução do Problema de Congestionamento Urbano utilizando Restauração Inexata

R. Andreani
andreani@ime.unicamp.br

J.L. Chela
chela@ime.unicamp.br

A. Friedlander
friedlan@ime.unicamp.br

Instituto de Matemática, Estatística e Computação Científica-IMECC.
Universidade Estadual de Campinas (UNICAMP), Campinas-SP, Brasil.

## 1 Introdução

O trafego congestionado tem se tornado parte do dia-a-dia dos moradores das grandes cidades nos últimos anos. Com o crescimento descontrolado nem sempre é viável construir ruas ou avenidas para suprir o aumento do fluxo de veículos, sendo assim necessário impor condições para diminuí-lo em certas regiões.

Uma maneira de controlá-lo seria estabelecer impostos sobre alguns trajetos. Na economia de transportes o problema de determinar os imposto visando otimizar o tempo total do trajeto e o fluxo de veículos(evitando congestionamento) é conhecido como **Congestion Princing Problem** ver [4].

Este problema pode ser reformulado como um problema de programação matemática com restrições de equilíbrio (MPEC) ver [4].

O objetivo deste trabalho consiste em resolver este MPEC utilizando um algoritmo de restauração inexata. Estes algoritmos são métodos iterativos onde cada iteração é realizada em duas fases: uma de "Restauração", onde a viabilidade é melhorada e outra fase chamada de "Otimização" onde se reduz a função objetivo do problema do primeiro nível sobre um conjunto viável aproximado.

Na fase de *Otimização* usamos a direção do gradiente projetado da função lagrangeana sobre o plano tangente associado ao sistema K.K.T dado pelo problema do segundo nível. A fase de *Restauração* consiste em aplicar na resolução do problema de equilíbrio uma estratégia de projeção junto com quadrados mínimos do sistema K.K.T . Dado que estas estratégias geram sequências de pontos que se aproximam ao conjunto solução do problema de equilíbrio, a distância a este conjunto é usada como medida de progresso para a restauração inexata.

Mostramos a formulação do problema **Congestion Princing Problem** como um MPEC.

## 2 Problema de Congestionamento

**Notações**

Para ajudar na formulação do problema definimos:

i- $K$ : conjunto dos índices de nós da rede;

ii- $\aleph$ : conjunto dos índices dos links(arcos) pertencentes a rede;

iii- $v_i$ : fluxo total sobre o link(arco) i;

iv- $d_j$ : demanda de fluxo no destino $j$;

v- $A \in I\!R^{k\times n}$ matriz de restrições de fluxo, onde k é o número de nós da rede e n número de links;

vi- $s_i$ : é o custo em tempo de uma unidade de fluxo $v_i$ sobre o link(arco) $i$;

vii- $\beta_i$ : é o imposto(em minutos) cobrado sobre o link(arco) $i$;

viii- $Y$ : subconjunto de $\aleph$ que corresponde aos arcos que não estão sujeitos a impostos.

### 2.1 Formulação do Problema como um MPEC

O problema de congestionamento pode ser formulado como um problema de programação matemática com restrições de equlíbrio (MPEC) [4] da seguinte forma:

$$\min_{v,\beta} \langle s(v), v\rangle \ s.a$$
$$\beta_i = 0 \text{ se } i \in Y$$
$$\beta_i \geq 0 \text{ se } i \notin Y \qquad \textbf{MPECT}$$
$$\langle s(v) + \beta, v - z\rangle \leq 0 \ \forall z \in V$$

onde $V = \{z \in I\!R^n : Az = b \ e \ z \geq 0\}$, $s_i : I\!R^n \to I\!R$ é pelo menos duas vezes diferenciável, separável e monótona, para $i \in \{1, .., n\}$.

Se Y é vazio o problema é conhecido como First Best Toll Princing caso contrário o problema é chamada de Second Best toll princing [4].

Para deixar claro o problema consideremos o seguinte exemplo:

**Exemplo 2.1** *Consideremos uma rede com uma origem e um destino onde é necessário levar* 6 *unidades de fluxo da origem até o destino através de dois arcos, neste caso* $A = [1\ 1]$, $b = 6$, $s_1(v) = v_1^2$ *e* $s_2(v) = v_2^2$.

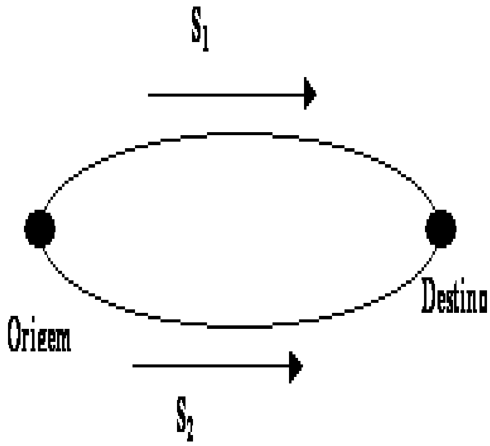


Figura 1:

A proposta deste trabalho, é resolver o problema **MPECT** utilizando uma técnica de restauração inexata [6].

# 3 Algoritmos

Consideremos o seguinte problema MPEC:

$$\min_{x,y} f(x, y) \ s.a$$
$$x \in X = \{x : a \leq x \leq b\} \ \subset I\!R^n \qquad \textbf{MPEC}$$
$$\langle G(x, y), y - z\rangle \leq 0 \ \forall z \in Y(x)$$

onde $Y(x) = \{y \in I\!R^m : h(x, y) = 0 \ e \ y \geq 0\}$, $f$ é diferenciável, $h_i(x, .) : I\!R^m \to I\!R^q$ é pelo menos duas vezes diferenciável e convexas para $i \in \{1, ..., q\}$ e para todo $x \in X$, $G : I\!R^n \times I\!R^m \to I\!R^m$ é diferenciável, $a_i > -\infty$ e $b_i < \infty$ para $i \in \{1, .., n\}$.

Definimos o operador

$$C(x, y, \lambda, z) = \begin{pmatrix} G(x, y) + \nabla^t h_y(x, y)\lambda - z \\ h(x, y) \\ z_1 y_1 \\ \dots \\ z_m y_m \end{pmatrix}, \quad (1)$$

e

$$\Omega = \{(x, y, \lambda, z) \in I\!R^{n\times m\times q\times m} : x \in X, \ y, z \geq 0\} \quad (2)$$

## 3.1 Algoritmo I

Encontre $\eta > 0$, $r \in [0, 1)$, $\epsilon > 0$, $\beta > 0$, $\gamma_0 \in R^{n+q+m}$,$M > 0$, $\theta_{-1} \in (0, 1)$, $\delta_{min} > 0$, $\tau_1 > 0$, $\tau_2 > 0$, $s_0 \in \Omega$ e $\{\bar{\alpha}_k\}$ tal que $\sum_{k=1}^{\infty} \bar{\alpha}_k < \infty$.

**Passo 1:** " Parâmetros Iniciais. Defina

$$\theta_k^{min} = \min\{1, \theta_{k-1}, ..., \theta_{-1}\}. \quad (3)$$
$$\theta_k^{large} = \min\{1, \theta_k^{min} + \alpha_k\} \quad e \quad (4)$$
$$\theta_{k,-1} = \theta_k^{large}. \quad (5)$$

**Passo 2:** "Fase de Restauração" Seja $s_k = (x_k, y_k, \lambda_k, z_k)$ iteração corrente.

Calcule $C(s_k)$.

2.1 : Se $C(s_k) = 0$, escreva $\bar{s}_k = s_k$.

senão

Fase I: Calcule $\bar{y}_k$ usando Algoritmo II tal que $\|R(\bar{y}_k)\| \leq \epsilon$.

Fase II:

Fixando $(x_k, \bar{y}_k)$, encontre $(\bar{\lambda}_k, \bar{z}_k)$ satisfazendo

$$\|C(x_k, \bar{y}_k, \bar{\lambda}_k, \bar{z}_k)\| \leq r\|C(x_k, y_k, \lambda_k, z_k)\|, \quad (6)$$

onde $r \in (0, 1)$, $\beta > 0$ e não dependem de k.

escreva $\bar{s}_k = (x_k, \bar{y}_k, \bar{\lambda}_k, \bar{z}_k, )$.

**Passo 3:** " Direção de Cauchy " Calcule

$$d_{tan}^k = P_k[\bar{s}_k - \eta\nabla L(\bar{s}_k, \gamma_k)] - \bar{s}_k \quad (7)$$

onde $P_k[z]$ é a projeção ortogonal de $z$ sobre $\pi_k = \{s \in \ \Omega : C'(\bar{s}_k)(s - \bar{s}_k) = 0\}$ e $L(s, \gamma) = f(x, y) + \langle\gamma, C(x, y, \lambda, z)\rangle$.

3.1 " Critério de Parada do Algoritmo "

Se $C(s_k) = C(\bar{s}_k) = 0$ e $d_{tan}^k = 0$ pare o algoritmo e retone $\bar{s}_k$ como solução.

Se $d^k_{tan} = 0$, calcule $\gamma_{k+1}$ tal que $\|\gamma_{k+1}\| \leq M$ e defina

$$s_{k+1} = \bar{s}_k \tag{8}$$
$$\theta_k = \theta_{k,-1} \tag{9}$$
$$Ared_k = (1-\theta_k)[\|C(s_k)\| - \|C(\bar{s}_k)\|]. \tag{10}$$

e pare a iteração $k$.

Senão

escolha $i = 0$, $\delta_{k0} \geq \delta_{min}$ e continue.

**Passo 4:** " Fase de Minimização "
Calcule $s_{ki} \in \pi_k$ usando Algorithm III tal que

$$\|s_{ki} - \bar{s}_k\| \leq \delta_{ki} \quad e \tag{11}$$
$$L(s_{ki}, \gamma_k) << L(s_k, \gamma_k). \tag{12}$$

onde $s_{ki} = (x_{ki}, y_{ki}, \lambda_{ki}, z_{ki})$.

**Passo 5:** " Trial Multiplicadores"

Calcule $\gamma^{trial}_{ki}$ tal que $\|\gamma^{trial}_{ki}\| \leq M$ .

**Passo 6:** " busca do $\theta$ ótimo "

Para todo $\theta \in [0,1]$ defina

$$Pred_{k,i}(\theta) = \theta[L(s_k,\gamma_k) - L(s_{k,i},\gamma_k) - \tag{13}$$
$$\langle C(\bar{s}_k), \gamma^{k,i}_{trial} - \gamma_k\rangle] \tag{14}$$
$$+(1-\theta)[\|C(s_k)\| - \|C(\bar{s}_k\|]. \tag{15}$$

Calcule $\theta_{k,i}$ o máximo $\theta \in [0, \theta_{k,i-1}]$ tal que

$$Pred_{ki}(\theta) \geq 0.5[\|C(s_k)\| - \|C(\bar{s}_k\|] \tag{16}$$

Defina $Pred_{ki} = Pred_{k,i}(\theta_{k,i})$

**Passo 7:** " Redução "

Seja

$$Ared_{k,i} = \theta_{k,i}[L(s_k,\gamma_k) - L(s_{k,i},\gamma^{k,i}_{trial})] + \tag{17}$$
$$(1-\theta_{k,i})[\|C(s_k)\| - \|C(s_{k,i})\|]. \tag{18}$$

Se

$$Ared_{k,i} \geq 0.1 Pred_{k,i} \tag{19}$$

Defina $s_{k+1} = s_{k,i}$, $\gamma_{k+1} = \gamma^{trial}_{k,i}$, $\theta_k = \theta_{k,i}$, $\delta_k = \delta_{k,i}$, $i_{aux}(k) = i$ e pare a iteração $k$.

Senão

escolha $\delta_{k,i+1} \in [0.1\delta_{k,i}, 0.9\delta_{k,i}]$, $i \longleftarrow i+1$ e volte ao passo 4.

## 3.2 Algoritmo II

Vamos introduzir os elementos e notações que serão usadas no decorrer do algoritmo. Dado $x_k$ e o correspondente $Y(x_k)$ do probelma **MPEC**, começando com $y_k \in I\!R^m$, duas sequências são geradas e denotadas por $\{\hat{y}_i\} \in I\!R^m$ e $\{\tilde{y}_i\} \in I\!R^m$. A determinação de um hiperplano separador é o passo chave do algoritmo. Consideremos a projeção ortonalgal sobre um conjunto S pela notação $P_S[.]$. Definos uma iteração projeção contínua por

$$R(y) = y - P_{\Omega(x_k)}[y - G(x_k, y)], \tag{20}$$

e, dado $\tilde{y}_i$, o ponto que define o semi espaço $H_i$ contendo o conjunto solução do $VIP[G(x_k,.), Y(x_k)]$, o conjunto $H_i$ é escrito como

$$H_i = \{z \in I\!R^m : \langle G(x_k, \tilde{y}_i), z - \tilde{y}_i\rangle \leq 0\}. \tag{21}$$

Dado $\gamma, \sigma \in (0,1)$, seja $p$ o inteiro não negativo tal que a desigualdade

$$\langle G(x_k, \hat{y}_i - \gamma^p R(\hat{y}_i)), R(\hat{y}_i)\rangle \geq \sigma\|R(\hat{y}_i)\|^2 \tag{22}$$

é satisfeita. Em seguida nós buscamos a viabilidade de $y_k$, calculando $\hat{y}_i = \bar{y}_k$ tal que (**??**) é satisfeita.

**Passo 1:**

Escolha $\gamma, \sigma \in (0,1)$ e $\hat{y}_0 = P_{\Omega(x_k)}[y_k]$. Tome $i = 0$.

Calcule $R(\hat{y}_0)$ e $R(y_k)$ usando (20).

**Passo 2:**

**Passo 2.1:** Se $i > 0$ calcule $R(\hat{y}_i)$ usando (20).

Se "$\|R(\hat{y}_i)\| \leq \epsilon$", pare o algoritmo e returne $\hat{y}_i$.

Caso contrário, calcule

$$\tilde{y}_i = \hat{y}_i - \eta_i R(\hat{y}_i), \tag{23}$$

onde $\eta_i = \gamma^{p_i}$, com $p_i$ sendo o menor $p$ não negativo tal que (22) é sastisfeito.

**Passo 2.2:** Calcule $\hat{y}_{i+1} = P_{\Omega(x_k)\cap H_i}[\hat{y}_i]$ onde $H_i$ é definido por (21), $i \longleftarrow i+1$ volte ao passo step 2.1.

## 3.3 Algoritmo III

**Passo 1:** Calcule $t^{break}_{k,i} = min\{1, \frac{\delta_{k,i}}{\|d^k_{tan}\|}\}$.

**Passo 2:** $Tome \longleftarrow t^{break}_{k,i}$.

**Passo 3:** Se

$$f(\bar{s}_k + t d^k_{tan}) \leq f(\bar{s}_k) + 0.1t\langle\nabla f(\bar{s}_k), d^k_{tan}\rangle, \tag{24}$$

defina $s_{k,i} \in \Omega$ such that $\|s_{k,i} - \bar{s}_k\| \leq \delta_{k,i}$ and

$$f(s_{k,i}) \leq \max\{f(\bar{s}_k + t d^k_{tan}), f(\bar{s}_k) - \tau_1\delta_{k,i}, f(\bar{s}_k) - \tau_2)\}. \tag{25}$$

Senão, escolha $t_{new} \in [0.1t, 0.9t]$, $\quad t \longleftarrow t_{new}$ e volte ao passo 3.

Para garantir a convergência do Algoritmo I , vamos supor que o problema MPEC satisfaz as seguintes hipóteses.

### 3.4 Hipóteses

Seja $(x, y^*(x), \lambda^*(x), z^*(x))$ uma solução do sistema K.K.T

$$C(x, y, \lambda, z) = 0 \quad \text{e} \quad (x, y, \lambda, z) \in \Omega. \tag{26}$$

H1- $w^t[\nabla G_y(x, y^*(x)) + \sum_i^m \nabla_y^2 h_i(x, y^*(x))\lambda_i]w > 0$, para todo $w \in N(h'(x, y^*(x)))$.

H2- Para cada $(x, y^*(x))$, temos que $(x, y^*(x))$ é regular, no sentido de que os conjunto de restrições ativas em $(x, y^*(x))$ são Linearmente Independentes.

H3- O conjunto $\Omega$ é convexo.

H4- Consideremos $s = (x, y, \lambda, z) \in \Omega$, então existe $L_2 > 0$, tal que para todo $s_1$ e $s_2 \in \Omega$,

$$\|C'(s_1) - C'(s_2)\| \leq L_2\|s_1 - s_2\|. \tag{27}$$

H5- Existe $L > 0$, tal que para todo $(x_1, y_1)$ e $(x_2, y_2) \in I\!R^n \times I\!R^m$

$$\|\nabla f(x_1, y_1) - \nabla f(x_2, y_2)\| \leq L\|(x_1 - x_2, y_1 - y_2)\|. \tag{28}$$

H6- Para todo $(x, y^*(x), \lambda(x), z(x))$ solução do sistema K.K.T temos,

$$y^*(x)_i + z(x)_i > 0 \;\forall i \in \{1, ..., m\}. \tag{29}$$

## Referências


[1] R. Andreani, J. M. Martínez and B. F. Svaiter. On the regularization of mixed complementarity problems. Numerical Functional Analysis and Optimization 21, pp. 589-600 (2000).

[2] J.F. Bard, Yo Ishizuka, K. Shimizu, Nondiferentiable and two level mathematical programming, Kluwer Academic Publishers(1997).

[3] S. Dempe. Foundations of Bilevel Programming, Kluwer Academic Publishers(2002).

[4] S. Lawphongpanich and D.W. Hearm, A MPEC approach to second-best toll princing, Math. Program, Ser B (2004)

[5] J. M. Martínez and E. A. Pilotta. Inexact restoration algorithms for constrained optimization. Journal of Optimization Theory and Applications 104, pp. 135-163 (2000).

[6] J.M. Martínez. *Inexact-Restoration Method with Lagrangian Tangent Decrease and New Merit Function for Nonlinear Propramming*, Journal of Opt. Theory and Applicat. vol 111, n.1, pp.39-58 (2001).

[7] M.V.Solodov and B.F.Svaiter, A New Projection Method for Variational Inequality Problems, SIAM J.Control Optim. Vol 37 n.3 pp 765-776 (1999).