

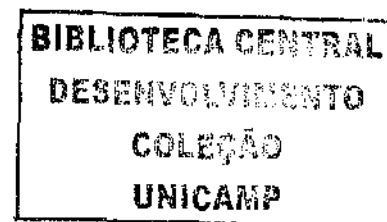
Universidade Estadual de Campinas
Instituto de Matemática, Estatística e Computação Científica
Departamento de Matemática Aplicada

**Resolução do Problema de Programação Matemática com
Restrições de Equilíbrio usando Restauração Inexata**

João Luiz Chela

Orientadora: Profa. Dra. Ana Friedlander

08 de março de 2006



UNIDADE	BE
Nº CHAMADA	11 UNICAMP
V	EX
TOMBO BC/	68295
PROC.	16. 123-06
C	☐
D	☒
PREÇO	11,00
DATA	04/5/06

Bib ID 378517

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Miriam Cristina Alves – CRB8a / 859

Chela, João Luiz .

C417r Resolução do problema de programação matemática com restrições de equilíbrio usando restauração inexata / João Luiz Chela -- Campinas, [S.P. :s.n.], 2006.

Orientadores : Ana Friedlander; Roberto Andreani.

Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Programação (Matemática). 2. Otimização matemática. 3. Transporte urbano. I. Friedlander, Ana. II. Andreani, Roberto. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Título em inglês: Resolution of mathematical program with equilibrium constraints using inexact restauration.

Palavras-chave em inglês (Keywords): 1. Mathematical program. 2. Mathematical optimization. 3. Urban transportation.

Área de concentração: Otimização.

Titulação: Doutor em Matemática

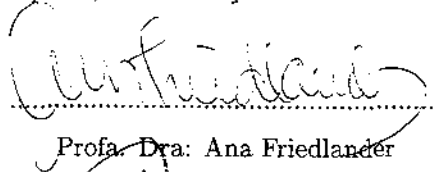
Banca examinadora: Prof. Dr. Ana Friedlander (IMECC-UNICAMP)
Profª. Dra. Susana Scheimberg de Makler (Coppe - UFRJ)
Prof. Dr. Nelson Maculan Filho (PESC/COPPE -UFRJ)
Prof. Dr. Alfredo Iusem (IMPA)
Profª. Dra. Sandra Augusta Santos (IMECC-UNICAMP)

Data da defesa: 08/03/2006

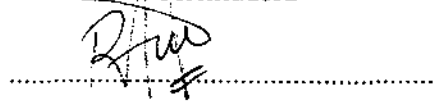
Resolução do Problema de Programação Matemática com Restrições de Equilíbrio usando Restauração Inexata

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por João Luiz Chela e aprovada pela comissão julgadora.

Campinas, 08 de março de 2006



Profa. Dra: Ana Friedlander
Orientadora



Prof. Dr. Roberto Andreani
Co-orientador

Banca Examinadora

Profa. Dra. Ana Friedlander (IMECC/UNICAMP) (Orientadora)

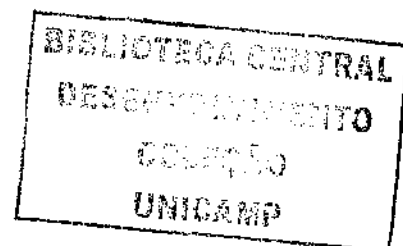
Profa. Dra. Susana Scheimberg de Makler (COPPE - UFRJ)

Prof. Dr. Nelson Maculan Filho (PESC/COPPE/UFRJ)

Prof. Dr. Alfredo Iusem (IMPA)

Profa. Dra. Sandra Augusta Santos (IMECC/UNICAMP)

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do Título de DOUTOR em Matemática Aplicada.



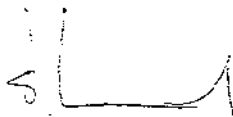
200609096

Tese de Doutorado defendida em 08 de março de 2006 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.



Prof. (a). Dr(a). ANA FRIEDLANDER DE MARTINEZ PEREZ



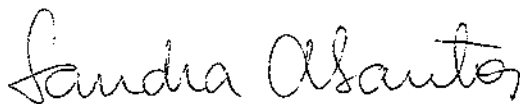
Prof. (a). Dr (a). SUSANA SCHEIMBERG DE MAKLER



Prof. (a). Dr (a). NELSON MACULAN FILHO



Prof. (a). Dr (a). ALFREDO NOEL TUSEM



Prof. (a) Dr. (a) SANDRA AUGUSTA SANTOS

Para Meus Pais, irmãos e Simone

Agradecimentos

À minha orientadora Ana Friedlander, pela dedicação e paciência.

Ao meu co-orientador e amigo Roberto Andreani, pela dedicação, amizade e pelos jogos do Corinthians que sofremos juntos.

Ao professor José Mário Martínez e família, pelas sugestões, apoio e almoços de Domingo.

À minha família, pelo carinho e compreensão.

A todos os professores do IMECC, em especial a Sandra Augusta Santos, Chetti, a todos meus colegas de pós graduação e colegas de trabalho do Banco Itaú S/A.

A todos os funcionários do IMECC, em especial a Fátima, Tânia, Cidinha e Ednaldo.

A FAPESP, pelo custeio dos meus estudos e auxílios financeiros.

Resumo

O Problema de Programação Matemática com Restrições de Equilíbrio (MPEC) consiste em um problema de otimização, onde a definição do conjunto viável inclui o conjunto de soluções de um problema de inequações variacionais. Também é denominada MPEC a reformulação do problema como um problema de otimização clássico, obtida substituindo o problema variacional pelo sistema de Karush-Kuhn-Tucker associado. O problema variacional é também chamado neste contexto problema do segundo nível.

A resolução do problema MPEC é mais difícil que a dos problemas clássicos de otimização. Esta dificuldade se deve basicamente à estrutura de dois níveis do problema MPEC. Existem diversos exemplos que mostram que a região viável pode não ser convexa e até mesmo desconexa. Mesmo no caso em que a trajetória de soluções dos problemas do segundo nível pode ser expressa como uma função dos parâmetros, a função objetivo do primeiro nível pode ser não diferenciável.

Neste trabalho, propomos uma nova abordagem para resolver problemas de Programação Matemática com Restrições de Equilíbrio. Esta abordagem permite que o problema do segundo nível seja resolvido diretamente, sem reformulações nem uso de técnicas não diferenciáveis. Para isso, utilizamos um Algoritmo de Restauração Inexata baseado no trabalho de Martínez em [50].

Apresentamos resultados teóricos e experimentos numéricos, incluindo aplicações.

Palavras chave. Programação Matemática com Restrições de Equilíbrio (MPEC); Restauração Inexata; Inequações Variacionais; Problema de Tráfego Urbano.

Abstract

A Mathematical Program with Equilibrium Constraints (MPEC) is an optimization problem, where part of the variables are constrained to be solutions of a variational inequality problem parameterized by the other variables. The reformulation of a MPEC, as a classical optimization problem, replacing the variational inequality problem by corresponding the K.K.T system, is also called MPEC. In this context the variational inequality problem is also called the second level problem.

MPEC problems are harder to solve than classical optimization problems due to their two-level structure. These problems are non-convex, and the feasible region can even be a disconnected one.

The objective function of the first level is in general non-differentiable, even in the case where the second level solutions can be expressed as a function of the parameters.

In this work, to solve Mathematical Programming Problems we use an Algorithm of Inexact Restoration based in the work of Martínez in [50]. This approach allows to treat the second level problem design without reformulation and we do not need any special algorithm designed for non-differentiable optimization.

We present theoretical results and numerical experiments, including an application in urban traffic problems.

Key words. Mathematical Program with Equilibrium Constraints (MPEC); Inexact Restoration; Variational Inequalities; Urban Traffic Problem.

Notação

$x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$	vetor no espaço n-dimensional
$\ x\ $	norma do vetor x
$A \in \mathbb{R}^{m \times n}$	matriz de dimensão $m \times n$
$A^T \in \mathbb{R}^{n \times m}$	transposta da matriz A
$ A $	norma da matriz A
$f : \mathbb{R}^n \rightarrow \mathbb{R}$	funções com valores reais
$\nabla f(x) \in \mathbb{R}^n$	gradiente da função f
$\nabla_{x_i, \dots, x_j} f(x) \in \mathbb{R}^{j-i+1}$	gradiente da função f , em relação às variáveis $x_i, \dots, x_j$
$\nabla^2 f(x) \in \mathbb{R}^{n \times n}$	matriz hessiana da função f
$h : \mathbb{R}^n \rightarrow \mathbb{R}^m$	funções com valores vetoriais
$h'(x) \in \mathbb{R}^{m \times n}$	matriz jacobiana da função h
$h'_{x_i, \dots, x_j}(x) \in \mathbb{R}^{m \times (j-i+1)}$	matriz jacobiana da função h em relação às variáveis $x_i, \dots, x_j$
$\nabla h(x) \in \mathbb{R}^{n \times m}$	matriz transposta da jacobiana de h
$x \geq 0$	$x_i \geq 0$ para todo $i = 1, \dots, n$
argmin	argumento que minimiza

$V(x^*, \varepsilon) = \{x \in X \mid \|x - x^*\| < \varepsilon\}$ vizinhança aberta de raio ε em torno do ponto x^*

$Nu(A) = \{x \in \mathbb{R}^n \mid Ax = 0\}$ Núcleo de $A \in \mathbb{R}^{m \times n}$

Conteúdo

Introdução	1
1 Preliminares	3
1.1 Programação Matemática com Restrições de Equilíbrio (MPEC)	3
1.1.1 Exemplos	7
1.1.2 Métodos de Resolução	9
1.2 Inequações Variacionais	10
1.2.1 Operadores monótonos e projeções	12
1.2.2 Métodos de resolução do VIP	14
2 Restauração Inexata Aplicada ao Problema MPEC	23
2.1 Definição do Problema	23
2.2 Algoritmo de Restauração Inexata para o Problema MPEC	25

2.2.1	Algoritmo 2.1	25
2.3	Hipóteses sobre o problema	29
2.4	Boa definição do algoritmo	31
2.5	Convergência do Algoritmo	35
2.5.1	Condições suficientes para convergência a pontos AGP	38
3	Experimentos Numéricos e Aplicações	42
3.1	Experimentos Numéricos	42
3.1.1	Algoritmos Implementados	43
3.1.2	Problemas Teste	44
3.1.3	Problemas de Estrutura de Redes	54
3.1.4	Custo de Transporte	56
4	Aplicação ao Problema de Tráfego Urbano	63
4.1	Introdução	63
4.2	Descrição do Problema	65
4.2.1	O Problema SBTPP com Demanda Elástica	65
4.2.2	Formulação MPEC	67

4.2.3	Sensibilidade ao Pedágio	72
4.2.4	Modelagem do Problema SBTPPF	74
4.3	Resolução via Algoritmo de Restauração Inexata	75
4.4	Resultados Numéricos	76
5	Conclusões e trabalho futuro	86
6	Apêndice	90
6.1	Limitações para o Erro	90
	Bibliografia	98

Introdução

Em diversas áreas de estudo, como na teoria de jogos, na engenharia de estruturas ou na otimização de desenhos de processos químicos e economia, surgem problemas envolvendo decisões hierárquicas. Eles podem ser modelados como problemas de Programação Matemática com Restrições de Equilíbrio (MPEC), em outras palavras, problemas de otimização cujo conjunto de restrições engloba um problema de inequações variacionais.

O problema MPEC é uma generalização do problema de programação matemática em dois níveis, onde o problema do segundo nível é um problema de otimização. Os problemas min-max, de otimização bilinear e multi-objetivo, podem ser formulados como problemas de programação matemática em dois níveis.

A resolução do problema MPEC é mais difícil que a dos problemas clássicos de otimização. Esta dificuldade se deve basicamente à estrutura de dois níveis do problema MPEC. Existem diversos exemplos que mostram que a região viável pode não ser convexa e até mesmo desconexa, e o problema em geral é não diferenciável.

As dificuldades citadas impedem a utilização de técnicas de resolução clássicas de programação matemática. Por causa disto diversos pesquisadores desenvolveram algoritmos especializados para resolver casos particulares de problemas MPEC, muitas

vezes provenientes de aplicações específicas.

Uma outra abordagem bastante utilizada é a reformulação como um problema de otimização que consiste em substituir o segundo nível pelo sistema Karush-Kuhn-Tucker (KKT) associado.

Um enfoque diferente foi apresentado em [18] para resolver o problema de programação em dois níveis e nesta tese estendemos os resultados obtidos e apresentamos um algoritmo para tratar o problema MPEC. A principal vantagem deste enfoque é a de permitir que o problema do segundo nível seja resolvido diretamente, sem reformulação, o que permite aproveitar melhor a sua estrutura em dois níveis. Também não precisamos usar técnicas não diferenciáveis.

A ferramenta essencial é o Algoritmo de Restauração Inexata proposto por Martínez ([50]). Este método, desenvolvido para resolver problemas clássicos de otimização, tem a característica de tratar a viabilidade e a otimalidade em diferentes fases a cada iteração, permitindo assim, que a estrutura de cada problema particular seja melhor explorada.

Organizamos o trabalho da seguinte forma: no Capítulo 1, descrevemos o problema MPEC e introduzimos algumas definições. No Capítulo 2, propomos a nova abordagem para a resolução do MPEC e apresentamos resultados de convergência. No Capítulo 3, apresentamos alguns testes numéricos para validar o algoritmo proposto. No Capítulo 4 aplicamos esta abordagem na resolução de alguns problemas de tráfego urbano. Finalmente, as conclusões e propostas de trabalho futuro estão apresentadas no Capítulo 5.

Capítulo 1

Preliminares

1.1 Programação Matemática com Restrições de Equilíbrio (MPEC)

O problema de Programação Matemática com Restrições de Equilíbrio consiste em:

$$\begin{aligned} & \underset{x,y}{\text{Minimizar}} && f(x,y) \\ & \text{s.a} && (x,y) \in \Omega, \\ & && \langle G(x,y), y - z \rangle \leq 0 \quad \forall z \in D(x), \\ & && y \in D(x), \end{aligned} \tag{1.1}$$

onde $\Omega \equiv X \times Y \subset \mathbb{R}^n \times \mathbb{R}^m$, $D(x) = \{y \in \mathbb{R}^m : h(x,y) = 0 \text{ e } g(x,y) \geq 0\}$, $f : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$, $h_i(x, \cdot) : \mathbb{R}^m \rightarrow \mathbb{R}$ são afins para $i \in \{1, \dots, q\}$ para todo $x \in X$, $g_i(x, \cdot) : \mathbb{R}^m \rightarrow \mathbb{R}$ são côncavas para $i \in \{1, \dots, l\}$ para todo $x \in X$ e $G : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$.

Nesta tese vamos nos referir a este problema como **MPEC**, denominação motivada

pelo inglês “*Mathematical Program with Equilibrium Constraints*”.

Consideremos os seguintes conjuntos:

Conjunto viável do segundo nível: Dado $x \in X$,

$$D(x) = \{y \in \mathbb{R}^m : h(x, y) = 0 \text{ e } g(x, y) \geq 0\}. \quad (1.2)$$

Conjunto de soluções do segundo nível: Dado $x \in X$,

$$M(x) = \{y \in D(x) : \langle G(x, y), y - z \rangle \leq 0 \quad \forall z \in D(x)\}. \quad (1.3)$$

Região Induzida:

Seja $x \in \mathbb{R}^n$, definimos

$$RI = \{(x, y) \in \mathbb{R}^{n \times m} : (x, y) \in \Omega \text{ e } y \in M(x)\}. \quad (1.4)$$

A região induzida é a região viável do problema MPEC.

O problema de otimização

$$\underset{x, y}{\text{Minimizar}} \quad f(x, y)$$

onde $(x, y) \in \Omega$ e y é solução do problema de inequações variacionais parametrizado por x , é denominado problema do primeiro nível. O problema

$$y \in D(x) \text{ e } \langle G(x, y), y - z \rangle \leq 0 \quad \forall z \in D(x) \quad (1.5)$$

será denotado por $VIP(G(x, \cdot), D(x))$ e será chamado de problema do segundo nível.

Um dos primeiros trabalhos a tratar o problema MPEC foi o de Marcotte [48] em 1985, no entanto o termo MPEC foi introduzido em 1988 por Harker e Pang [36]. Casos

particulares do problema MPEC já foram tratados desde 1973 por Bracken e McGill [12]. A importância prática é relevante, tendo em vista que o seu campo de aplicação é muito amplo, como mencionado na introdução. Ver [7, 13, 23, 38, 39, 40, 41, 45, 48]. Nos livros [25, 46] são estudados em profundidade os problemas de programação matemática em dois níveis e o MPEC, respectivamente.

Em vários textos o problema $VIP(G(x, \cdot), D(x))$ é definido sem assumir que a região $D(x)$ é convexa. Acreditamos que não é correto estender essa definição para este caso, já que é fácil mostrar exemplos de problemas de minimização em regiões não convexas, onde o minimizador global não é solução do $VIP(G(x, \cdot), D(x))$ natural associado. Portanto, para o caso em que $D(x)$ não é convexa, o problema que nos interessa estudar é o problema de programação matemática cujas restrições incluem o sistema KKT associado a $D(x)$, e que em alguns textos é chamado problema de programação matemática com restrições de complementaridade (MPCC). Ver [43, 73]. Portanto, não incluiremos este problema entre os denominados MPEC.

Na programação matemática com restrições de equilíbrio podemos identificar classes específicas de problemas MPEC, entre estas merece destaque a Programação Matemática em Dois Níveis [9, 17, 25, 26, 27, 28].

Programação Matemática em Dois Níveis

O problema de Programação Matemática em Dois Níveis (BP) é o principal caso especial do problema MPEC e pode ser expresso da seguinte forma:

$$\begin{aligned}
 & \underset{x,y}{\text{Minimizar}} && f(x,y) \\
 & \text{s.a} && (x,y) \in \Omega \\
 & && y \text{ solução de} \\
 & \underset{y}{\text{Minimizar}} && F(x,y) \\
 & \text{s. a} && y \in D(x)
 \end{aligned} \tag{1.6}$$

onde $\Omega \subset \mathbb{R}^n \times \mathbb{R}^m$, $D(x) = \{y \in \mathbb{R}^m : h(x,y) = 0, \text{ e } g(x,y) \geq 0\}$, $f : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$, $F(x, \cdot) : \mathbb{R}^m \rightarrow \mathbb{R}$, $h_i(x, \cdot) : \mathbb{R}^m \rightarrow \mathbb{R}$ para $i \in \{1, \dots, q\}$ para todo $x \in X$, $g_i(x, \cdot) : \mathbb{R}^m \rightarrow \mathbb{R}$ para $i \in \{1, \dots, l\}$ para todo $x \in X$. Esta nomenclatura surgiu somente em 1977 com Candler e Norton [17]. Se $D(x)$ e $F(x, \cdot)$ são convexas para todo $x \in X$, e $\nabla_y F(x, y)$ é contínua e diferenciável, resolver o problema (1.6) é equivalente a resolver o problema MPEC substituindo $G(x, y) = \nabla_y F(x, y)$.

Um problema que surge em diversas áreas de aplicação, que envolve uma estrutura hierárquica (em níveis) e que em vários textos é considerado como um problema de otimização em dois níveis é o Problema Stackelberg [65]. Ele consiste em

$$\begin{aligned}
 & \underset{x}{\text{Minimizar}} && f(x,y) \\
 & \text{s.a} && x \in X \\
 & && y \text{ solução de} \\
 & \underset{y}{\text{Minimizar}} && F(x,y) \\
 & \text{s.a} && y \in D(x)
 \end{aligned} \tag{1.7}$$

onde $X \subset \mathbb{R}^n$, $D(x) = \{y \in \mathbb{R}^m : h(x,y) = 0, \text{ e } g(x,y) \geq 0\}$, $f : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$, $F(x, \cdot) : \mathbb{R}^m \rightarrow \mathbb{R}$, $h_i(x, \cdot) : \mathbb{R}^m \rightarrow \mathbb{R}^q$ para $i \in \{1, \dots, q\}$ para todo $x \in X$, $g_i(x, \cdot) : \mathbb{R}^m \rightarrow \mathbb{R}^l$

para $i \in \{1, \dots, l\}$ e para todo $x \in X$. Este problema não está bem definido se a solução do problema do segundo nível não é única. Isto não acontece com o problema em dois níveis definido acima, que sempre está bem definido. A razão de definir (1.7) desta forma se deve ao desejo de modelar situações onde não há cooperação entre os dois níveis.

1.1.1 Exemplos

Nos exemplos que seguem mostramos que a região induzida pode não ser convexa e até mesmo desconexa.

Exemplo 1.1. Infinitas Soluções no Problema do Segundo Nível

$$f(x, y) = x^2 + y^2$$

$$X = \{x \in \mathbb{R} : -1 \leq x \leq 1\}$$

$$G(x, y) = -x$$

$$D(x) = \{y \in \mathbb{R} : 0 \leq y \leq 1\}.$$

A solução do segundo nível é dada por:

$$M(x) = \begin{cases} [0, 1] & \text{se } x = 0 \\ 0 & \text{se } -1 \leq x < 0 \\ 1 & \text{se } 0 < x \leq 1 \end{cases}$$

A solução global é $(0, 0)$.

Ver figura 1.3.

Exemplo 1.2. *Conexidade* [67]

1- Região Viável do MPEC Conexa

$$f(x, y) = x - 2y$$

$$\Omega = \{(x, y) \in \mathbb{R}^2 : -1 \leq x \leq 1\}$$

$$G(x, y) = 1$$

$$D(x) = \{y \in \mathbb{R} : -1 \leq x + y \leq 1, \quad -1 \leq x - y \leq 1\}$$

Ver figura 1.4.

2- Região Viável do MPEC Desconexa

$$f(x, y) = x - 2y$$

$$\Omega = \{(x, y) \in \mathbb{R}^2 : -1 \leq x \leq 1 \text{ e } y \geq -\frac{1}{2}\}$$

$$G(x, y) = 1$$

$$D(x) = \{y \in \mathbb{R} : -1 \leq x + y \leq 1, \quad -1 \leq x - y \leq 1\}$$

Ver figura 1.5.

Exemplo 1.3. Minimizadores Locais [65, 18]

$$f(x, y) = (x - 5)^2 + (2y + 1)^2$$

$$\Omega = \{(x, y) \in \mathbb{R}^2 : x \geq 0\}$$

$$G(x, y) = 2(y - 1) - 1.5x$$

$$D(x) = \{y \in \mathbb{R} : -3x + y \leq 3, x + y \leq 7, x - 0.5y \leq 4 \text{ e } y \geq 0\}.$$

A solução global é $(1, 0)$ e $(5, 2)$ é um minimizador local.

1.1.2 Métodos de Resolução

As dificuldades ilustradas nos exemplos impedem a utilização de técnicas de resolução clássicas de programação matemática. Para superar esta deficiência muitos algoritmos específicos foram desenvolvidos para resolver o problema MPEC para casos particulares.

Uma abordagem bastante utilizada é substituir o problema do segundo nível pelo sistema KKT associado, tornando assim o problema MPEC um problema de otimização clássico:

$$\begin{aligned} & \underset{x, y, \mu, \gamma}{\text{Minimizar}} && f(x, y) \\ & \text{s.a} && C(x, y, \mu, \gamma) = 0 \\ & && (x, y) \in \Omega \\ & && g(x, y) \geq 0 \\ & && \gamma \geq 0. \end{aligned} \tag{1.8}$$

onde $\Omega \subset \mathbb{R}^n \times \mathbb{R}^m$, $y \in \mathbb{R}^m$, $\mu \in \mathbb{R}^q$, $\gamma \in \mathbb{R}^l$, e

$$C(x, y, \mu, \gamma) = \begin{pmatrix} G(x, y) + \sum_{i=1}^q \nabla_y h_i(x, y) \mu_i - \sum_{i=1}^l \nabla_y g_i(x, y) \gamma_i \\ h(x, y) \\ \gamma_1 g_1(x, y) \\ \vdots \\ \gamma_l g_l(x, y) \end{pmatrix} \tag{1.9}$$

A equivalência entre os problemas MPEC e (1.8) só está garantida quando a

região viável do problema do segundo nível é convexa e o operador $G(x, \cdot)$ é monótono para todo x . As restrições do problema reformulado implicam na singularidade de $C'_{x,y,\mu,\gamma}(x, y(x), \mu(x), \gamma(x))$ a menos que se assumam hipóteses mais fortes sobre o problema. Por outro lado, as soluções do problema reformulado não verificam nenhuma das qualificações clássicas.

Existem muitos artigos onde são apresentadas propostas para resolver o MPEC e BP, para casos particulares. Ver em [8, 9, 12, 14, 16, 17, 22, 30, 47, 48, 49, 55, 58, 59, 65, 67, 68, 69, 70, 62, 63, 64, 73, 36, 25, 46] e nas referências citadas nestes artigos.

No caso de programação em dois níveis, em [18] propõe-se uma estratégia de resolução baseada em um método de Restauração Inexata [50], que permite resolver diretamente o problema, sem reformulação e sem a necessidade de recorrer a técnicas de otimização não diferenciável. Nesta tese queremos estender este estudo para problemas MPEC.

1.2 Inequações Variacionais

O Problema de Inequações Variacionais (VIP) consiste em encontrar $x^* \in Y \subset \mathbb{R}^n$ tal que

$$\langle G(x^*), z - x^* \rangle \geq 0 \quad \forall z \in Y \quad (1.10)$$

onde $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ e Y é um subconjunto de $\mathbb{R}^n$ convexo e fechado.

Geometricamente se x^* é uma solução do VIP então $-G(x^*)$ pertence ao cone normal a Y em x^* , denotado por $N(x^*)$, onde

$$N(x^*) = \begin{cases} \{z : (y - x^*)^T z \leq 0 \quad \forall y \in Y\} & \text{se } x^* \in Y \\ \emptyset & \text{caso contrário} \end{cases} \quad (1.11)$$

Se considerarmos $Y = \{x \in \mathbb{R}^2 : g^1(x) \leq 0 \text{ e } g^2(x) \leq 0\}$, a interpretação geométrica de $N(x^*)$ é dada na figura 1.1.

Casos particulares do VIP:

(a) Problema de Complementaridade Não Linear(NCP).

Consiste em encontrar x^* tal que:

$$x^* \geq 0 \quad G(x^*) \geq 0 \quad \text{e} \quad G(x^*)^T x^* = 0, \quad (1.12)$$

onde $Y = \mathbb{R}_+^n$. Se considerarmos $G(x) = Mx + q$ onde $M \in \mathbb{R}^{n \times n}$, q (fixo) $\in \mathbb{R}^n$, trata-se de um problema de complementaridade linear(LCP).

Se $Y = \{x \in \mathbb{R}^n : a \leq x \leq b\}$, o problema é chamado de complementaridade mista (MPC).

(b) Resolução de sistemas não lineares. Neste caso o VIP é usado como ferramenta de resolução de sistemas não lineares monótonos, pois se considerarmos $Y = \mathbb{R}^n$, resolver o *VIP* é equivalente a encontrar $x \in \mathbb{R}^n$ tal que $G(x) = 0$.

(c) Problemas de Otimização. Se considerarmos $G(x) = \nabla f(x)$, resolver o VIP consiste em encontrar pontos que satisfaçam as condições necessárias de otimalidade de primeira ordem do seguinte problema de otimização

$$\underset{x \in Y}{\text{Minimizar}} \quad f(x)$$

onde $Y \subset \mathbb{R}^n$ convexo, $f : \mathbb{R}^n \rightarrow \mathbb{R}$, f é contínua, e diferenciável. Em [35] podemos encontrar outros casos particulares do VIP.

1.2.1 Operadores monótonos e projeções

As definições, teoremas e proposições citados nesta seção podem ser encontrados em [35, 60, 20].

Seja $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

(a) G é monótono se

$$[G(x) - G(y)]^T(x - y) \geq 0 \quad \forall x, y \in \mathbb{R}^n.$$

Equivale a dizer que o operador ∇G é semi-definido positivo.

(b) G é pseudomonótono se

$$G(y)^T(x - y) \geq 0 \implies G(x)^T(x - y) \geq 0 \quad \forall x, y \in \mathbb{R}^n.$$

(c) G é fortemente monótono se existe $\gamma > 0$ tal que

$$[G(x) - G(y)]^T(x - y) \geq \gamma \|x - y\|^2 \quad \forall x, y \in \mathbb{R}^n.$$

Seja $Y \subset \mathbb{R}^n$ um conjunto convexo, fechado e não vazio. Definimos a projeção de $x \in \mathbb{R}^n$ sobre Y como

$$P_Y(x) = \operatorname{argmin}_{y \in Y} \|x - y\|$$

onde $P_Y : \mathbb{R}^n \rightarrow Y$.

Proposição 1.1. Seja Y um conjunto convexo, fechado e não vazio. Então, dado $z \in \mathbb{R}^n$, temos $\langle z - P_Y(z), x - P_Y(z) \rangle \leq 0 \quad \forall x \in Y$. Ver *Proposição 1.3* em [20].

Teorema 1.1. (Continuidade da projeção.) Seja Y um conjunto convexo, fechado e não vazio em $\mathbb{R}^n$. Então $\|P_Y(x) - P_Y(y)\| \leq \|x - y\| \quad \forall x, y \in \mathbb{R}^n$. Ver *Teorema 1.3* em [20].

Proposição 1.2. (Unicidade da Projeção.) Dados $z \in \mathbb{R}^n$ e Y convexo, fechado e não vazio, então a projeção de z sobre Y ($P_Y(z)$) é única. Ver Proposição 1.4 em [20].

Sejam C_1 e C_2 conjuntos em $\mathbb{R}^n$ não vazios.

- (a) O hiperplano H separa C_1 e C_2 , se C_1 está contido em um dos semi-espacos fechados caracterizado por H , e C_2 está contido no outro semi-espaço.
- (b) H separa estritamente C_1 e C_2 se $C_1 \cap H = \emptyset$ e $C_2 \cap H = \emptyset$.
- (c) H separa C_1 e C_2 fortemente se existe um $\epsilon > 0$ tal que $C_1 + \epsilon B$ está contido em um semi-espaço aberto associado a H e $C_2 + \epsilon B$ está contido no semi-espaço aberto oposto, onde B é a bola euclidiana de raio um.

Teorema 1.2. Sejam C_1 e C_2 conjuntos em $\mathbb{R}^n$ não vazios. Existe um hiperplano separando propriamente C_1 e C_2 se e somente se existe um vetor b tal que

$$(a) \inf\{\langle x, b \rangle | x \in C_1\} \geq \sup\{\langle x, b \rangle | x \in C_2\}$$

$$(b) \sup\{\langle x, b \rangle | x \in C_1\} > \inf\{\langle x, b \rangle | x \in C_2\}$$

e existe um hiperplano separando C_1 e C_2 fortemente se e somente se existe um vetor b tal que

$$(c) \inf\{\langle x, b \rangle | x \in C_1\} > \sup\{\langle x, b \rangle | x \in C_2\}$$

Prova: Ver [60].

O fato de dois conjuntos poderem ou não ser separados é uma questão de existência e não é nenhuma surpresa que nas aplicações da teoria da separação ocorram demonstrações de vários teoremas de existência.

Tipicamente, precisamos de um vetor b com certas propriedades em um hiperplano, e que seja capaz de separar dois conjuntos convexos. O vetor b em questão é o vetor diretor do hiperplano separador de C_1 e C_2 .

1.2.2 Métodos de resolução do VIP

Sejam Y um subconjunto não vazio convexo e fechado de $\mathbb{R}^n$ e $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ contínua. É fácil ver que x^* é solução do $VIP(G, Y)$ se e somente se $r(x^*) = x^* - P_Y[x^* - G(x^*)] = 0$. Ver [35].

Devido à grande importância prática, muitos autores têm estudado intensivamente os problemas VIP, dentre estes podemos destacar [1, 2, 3, 4, 6, 19, 20, 31, 42, 66, 29] e referências internas.

Em [20] foi proposta uma estratégia de resolução utilizando a função de Fischer-Burmeister Penalizada [21], para o caso em que G é fortemente monótona. Uma vantagem desta estratégia é o fato de não ser necessária a convexidade das funções que definem as restrições. Neste mesmo trabalho foi proposta outra estratégia baseada em perturbações como as usadas nos métodos de tipo Ponto Proximal.

Outra técnica de resolução do VIP importante é baseada em projeções ortogonais, principalmente quando G e Y têm estruturas especiais [19, 42, 66].

Se G é contínua e monótona e $Y \in \mathbb{R}^n$ é convexo e fechado, um algoritmo baseado em projeções ortogonais, proposto por Solodov e Svaiter [66] é descrito a seguir :

Dados $x_o \in Y$, $\gamma \in (0, 1)$ e $\alpha \in (0, 1)$, se x_i foi calculado, os passos para obter x_{i+1} são

Algoritmo 1

Passo 1.(Critério de Parada)

Calcular $P_Y[x_i - G(x_i)]$

Se $r(x_i) = x_i - P_Y[x_i - G(x_i)] = 0$ (**Pare**).

Passo 2.(Busca Linear) Calcular $z_i = x_i - \eta_i r(x_i)$ onde $\eta_i = \gamma^{k_i}$ e k_i é o menor número inteiro k positivo que satisfaz

$$\langle G(x_i - \gamma^k r(x_i)), r(x_i) \rangle \geq \alpha \|r(x_i)\|^2 \quad (1.13)$$

Passo 3.(Atualização de x_i) $x_{i+1} = P_{Y \cap S_i}[x_i]$ onde $S_i = \{x \in \mathbb{R}^n : \langle G(z_i), x - z_i \rangle \leq 0\}$.

Este algoritmo gera uma sequência de pontos $\{x_i\}$ tal que o ponto x_{i+1} é a projeção ortogonal do ponto x_i sobre a interseção de Y com um hiperplano que separa x_i do conjunto de soluções do $VIP(G, Y)$.

Esta estratégia supõe que x_i é uma aproximação de uma solução do $VIP(G, Y)$. Então, calcula $P_Y[x_i - G(x_i)]$ e em seguida é feita uma busca linear entre x_i e $P_Y[x_i - G(x_i)]$ para obter z_i de modo que $r(x_i) = x_i - P_Y[x_i - G(x_i)]$ faça um ângulo agudo com $G(z_i)$, vetor diretor do hiperplano $H_i = \{x \in \mathbb{R}^n : \langle G(z_i), x - z_i \rangle = 0\}$ que é gerado de forma a separar x_i do conjunto de soluções do $VIP(G, Y)$.

Em seguida é computado x_{i+1} como a projeção de x_i sobre a interseção do hiperplano H_i e o conjunto Y . A figura 1.2 descreve este processo de projeção.

O próximo resultado mostra que a busca linear necessária nesta estratégia está bem

definida.

Lema 1.1. O hiperplano $H_i = \{x \in \mathbb{R}^n : \langle G(z_i), x - z_i \rangle = 0\}$ separa x_i do conjunto de soluções do $VIP(G, Y)$.

Prova: Sejam x_i e z_i como no processo descrito acima, e x^* uma solução do $VIP(G, Y)$. Como G é monótona então, de

$$\langle G(z_i) - G(x^*), z_i - x^* \rangle \geq 0 \quad (1.14)$$

resulta

$$\langle G(z_i), z_i - x^* \rangle \geq \langle G(x^*), z_i - x^* \rangle. \quad (1.15)$$

Como x^* é solução do $VIP(G, Y)$ então

$$\langle G(z_i), x^* - z_i \rangle \leq 0. \quad (1.16)$$

Suponhamos que x_i não é uma solução do $VIP(G, Y)$; como z_i foi obtido de forma que o ângulo entre $r(x_i)$ e $G(z_i)$ é agudo e pelo Algoritmo 1, $z_i = x_i - \eta_i r(x_i)$. Então,

$$\langle G(z_i), x_i - z_i \rangle > 0. \quad (1.17)$$

Por (1.16) e (1.17) o hiperplano H_i separa x_i do conjunto de soluções do $VIP(G, Y)$. ■

Os próximos resultados mostram a convergência da sequência $\{x_i\}$ gerada pelo Algoritmo 1, para uma solução do $VIP(G, Y)$.

Lema 1.2. Suponhamos que a busca linear do Algoritmo 1 está bem definida. Então

$$x_{i+1} = P_{Y \cap S_i}[x_i^*]$$

onde $x_i^* = P_{S_i}[x_i]$.

Prova: Ver Lema 2.2 [66]. ■

Teorema 1.3. Seja G contínua e monótona e suponhamos que o conjunto de soluções do $VIP(G, Y)$ é não vazio. Então a sequência $\{x_i\}$ gerada pelo Algoritmo 1 converge para a solução do $VIP(G, Y)$.

Prova: Ver Teorema 2.1 [66]. ■

Motivados por experimentos numéricos realizados em [66] os autores desta estratégia propuseram algumas alterações no Algoritmo 1 que influenciam no seu desempenho.

Uma alteração consiste em substituir a definição de $r(x)$ no Algoritmo 1 por $r(x, \mu) = x - P_Y[x - \mu G(x)]$.

O algoritmo modificado fica da seguinte forma:

Dados $x_0 \in Y$, $\gamma \in (0, 1)$, $\alpha \in (0, 1)$, $\eta_{-1} > 0$ e $\theta > 1$, os passos para obter x_{i+1} a partir de x_i são

Algoritmo 2

Passo 1.(Critério de Parada)

Calcular $P_Y[x_i - G(x_i)]$

Se $r(x_i) = x_i - P_Y[x_i - G(x_i)] = 0$ (**Pare**).

Computar $r(x_i, \mu_i)$ onde $\mu_i = \min\{\theta\eta_{i-1}, 1\}$.

Passo 2.(Busca Linear) Calcular $z_i = x_i - \eta_i r(x_i, \mu_i)$ onde $\eta_i = \gamma^{k_i} \mu_i$

onde k_i é o menor inteiro k que satisfaz

$$\langle G(x_i - \gamma^k \mu_i r(x_i, \mu_i)), r(x_i, \mu_i) \rangle \geq \alpha \|r(x_i, \mu_i)\|^2 \quad (1.18)$$

Passo 3.(Atualização de x_i)

$$x_{i+1} = P_{Y \cap S_i}[x_i]$$

onde $S_i = \{x \in \mathbb{R}^n : \langle G(z_i), x - z_i \rangle \leq 0\}$.

O próximo teorema é uma adaptação do Teorema 1.3 para este novo algoritmo.

Teorema 1.4. Se G é contínua, monótona e o conjunto de soluções do $VIP(G, Y)$ é não vazio, a sequência $\{x^k\}$ gerada pelo Algoritmo 2 converge à solução do $VIP(G, Y)$.

Prova: Ver Teorema 2.2 [66]. ■

É importante ressaltar que os resultados também são válidos quando G é pseudomonótona.

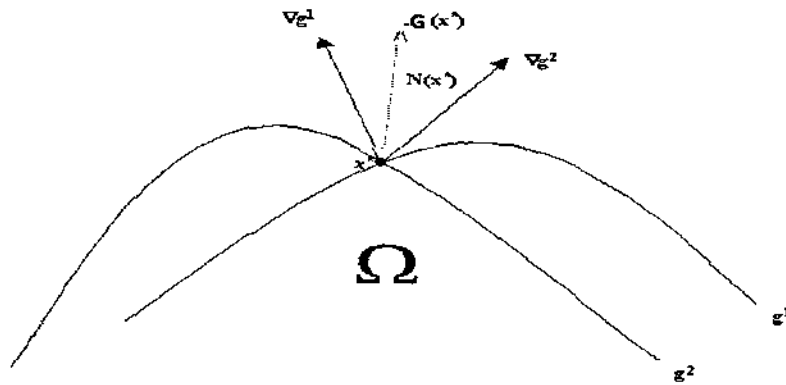


Figura 1.1: Cone Normal $N(x^*)$

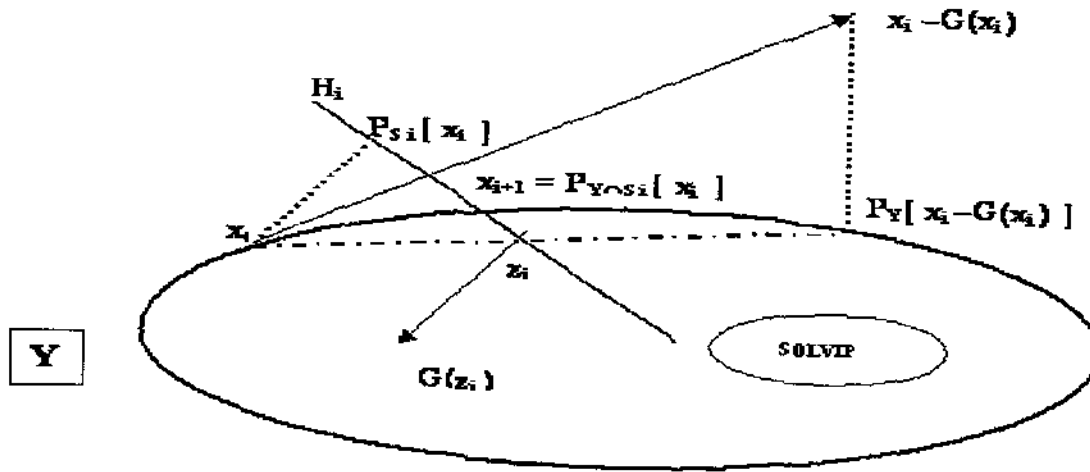


Figura 1.2: Ilustração para a Estratégia de Projeção

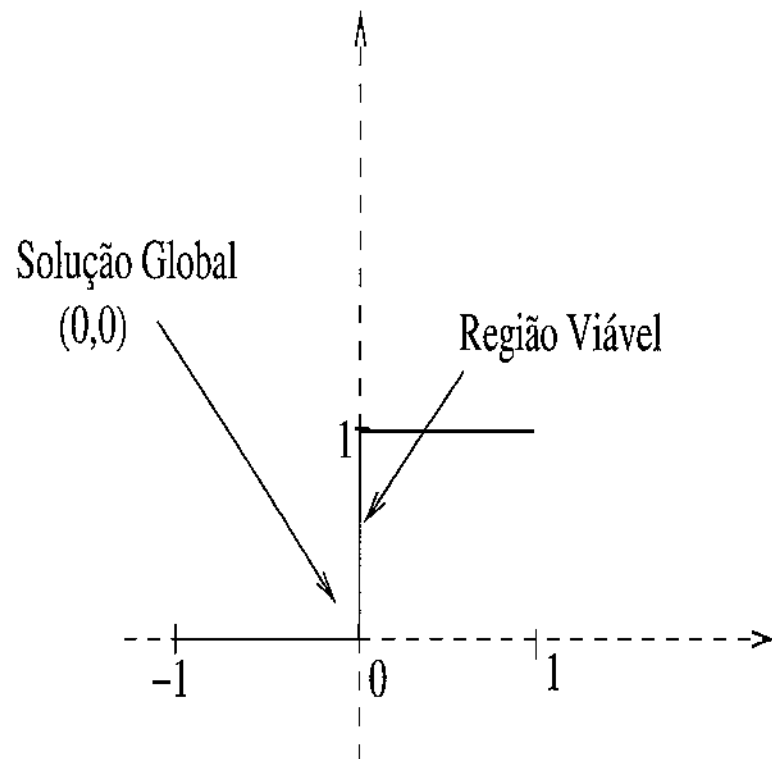


Figura 1.3: Ilustração para o Exemplo 1.1

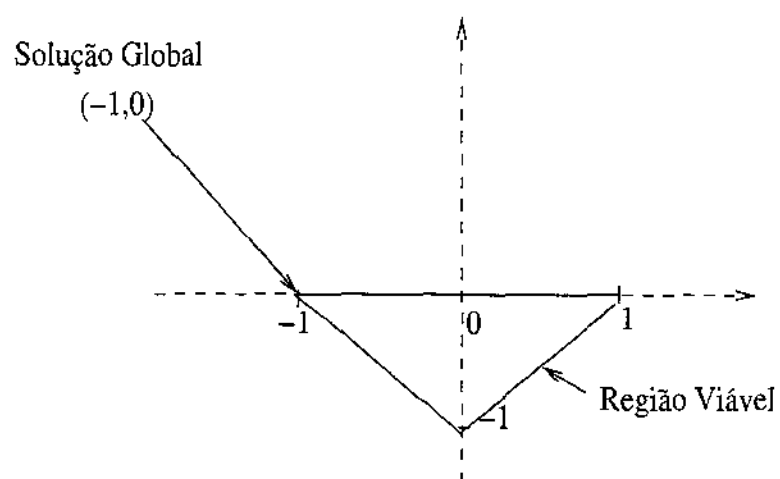


Figura 1.4: Ilustração para o Exemplo 1.2 Região Viável Conexa

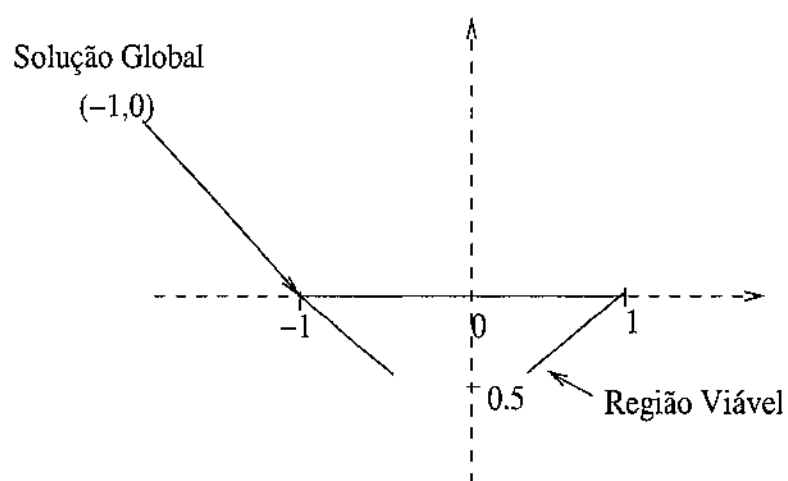


Figura 1.5: Ilustração para o Exemplo 1.2, Região Viável não Conexa

Capítulo 2

Restauração Inexata Aplicada ao Problema MPEC

2.1 Definição do Problema

Neste trabalho, estamos interessados em estudar o seguinte problema de Programação Matemática com Restrições de Equilíbrio.

$$\begin{aligned} & \underset{x,y}{\text{Minimizar}} && f(x,y) \\ & \text{s.a} && x \in X \\ & && y \text{ solução de} \\ & && y \in D(x) \\ & && \langle G(x,y), y - z \rangle \leq 0 \forall z \in D(x), \end{aligned} \tag{2.1}$$

onde $X \subset \mathbb{R}^n$ é um convexo, $D(x) = \{y \in \mathbb{R}^m : h(x,y) = 0 \text{ e } g(x,y) \geq 0\}$ é tal que para todo x , $D(x) \subset Y \subset \mathbb{R}^m$, Y convexo, $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ é diferenciável,

$h_i(x, \cdot) : \mathbb{R}^m \rightarrow \mathbb{R}$ são afins para $i \in \{1, \dots, q\}$ para todo $x \in X$, $g_i(x, \cdot) : \mathbb{R}^m \rightarrow \mathbb{R}$, são côncavas e duas vezes diferenciáveis para $i \in \{1, \dots, l\}$ e $G : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ é diferenciável.

Definimos,

$$C(x, y, \mu, \gamma) = \begin{pmatrix} G(x, y) + \sum_{i=1}^q \nabla_y h_i(x, y) \mu_i - \sum_{i=1}^l \nabla_y g_i(x, y) \gamma_i \\ h(x, y) \\ \gamma_1 g_1(x, y) \\ \vdots \\ \gamma_l g_l(x, y) \end{pmatrix} \quad (2.2)$$

e

$$L(x, y, \mu, \gamma, \lambda) = f(x, y) + C(x, y, \mu, \gamma)^T \lambda, \quad (2.3)$$

onde $(\mu, \gamma) \in \Delta \subset \mathbb{R}^q \times \mathbb{R}^l$, com Δ convexo, é o par de vetores multiplicadores de Lagrange do problema do segundo nível $VIP(G(x, \cdot), D(x))$ associado a $(x, y) \in \Omega \equiv X \times Y$. Temos que

$$\begin{aligned} C(x, y, \mu, \gamma) &= 0 \\ g(x, y) &\geq 0 \\ \gamma &\geq 0 \end{aligned} \quad (2.4)$$

é o sistema KKT associado ao problema do segundo nível, ou seja nesta situação deve ser considerado parametrizado por x . O vetor λ representa os multiplicadores de Lagrange relativo às igualdades do problema MPEC obtido substituindo o problema do segundo nível pelo sistema KKT (2.4).

2.2 Algoritmo de Restauração Inexata para o Problema MPEC

Consideremos a seguinte variação do algoritmo introduzido em [50] que gera uma sequência $s^k = (x^k, y^k, \mu^k, \gamma^k)$ para resolver o problema MPEC.

Dados $\eta > 0$, $r \in [0, 1)$, $\epsilon > 0$, $\lambda^0 \in \mathbb{R}^{m+q+l}$, $M > 0$, $\theta_{-1} \in (0, 1)$, $\delta_{min} > 0$, $\tau_1 > 0$, $\tau_2 > 0$, $s^0 \in \{(x, y, \mu, \gamma) \in \Omega \times \Delta : g(x, y) \geq 0, \gamma \geq 0\}$ e $\{w_k\}$ tal que $\sum_{k=1}^{\infty} w_k < \infty$. Assuma que para $k \in \{0, 1, 2, \dots\}$, $s^k = (x^k, y^k, \mu^k, \gamma^k) \in \Omega \times \Delta$, $\lambda^k \in \mathbb{R}^{m+q+l}$ e $\theta_{k-1}, \theta_{k-2}, \dots, \theta_0$ foram calculados. Então, os passos do Algoritmo para obter s^{k+1} são:

2.2.1 Algoritmo 2.1

Passo 1. Inicialização de parâmetros θ .

Defina

$$\theta_k^{min} = \min\{1, \theta_{k-1}, \dots, \theta_{-1}\}. \quad (2.5)$$

$$\theta_k^{large} = \min\{1, \theta_k^{min} + w_k\} \quad e \quad (2.6)$$

$$\theta_{k,-1} = \theta_k^{large}. \quad (2.7)$$

Passo 2. Fase de Restauração.

Se $C(s^k) = 0$, defina $z^k = s^k$. Senão, calcule $(\bar{y}, \bar{\mu}, \bar{\gamma}) \in Y \times \Delta$, uma solução inexata do problema $VIP(G(x^k, \cdot), D(x^k))$ tal que $z^k = (x^k, \bar{y}, \bar{\mu}, \bar{\gamma})$ verifique

$$\|C(z^k)\| \leq r\|C(s^k)\| \quad (2.8)$$

e

$$g(x^k, \bar{y}) \geq 0, \quad \bar{\gamma} \geq 0.$$

Passo 3. Direção de Cauchy.

Calcule

$$d_{tan}^k = P_k[z^k - \eta \nabla L(z^k, \lambda^k)] - z^k \quad (2.9)$$

onde $P_k[z]$ é a projeção ortogonal de z sobre $\pi_k = \{s : C'(z^k)(s - z^k) = 0\} \cap \{(x, y, \mu, \gamma) \in \Omega \times \Delta : g(x, y) \geq 0, \gamma \geq 0\}$. Se $C(s^k) = C(z^k) = 0$ e $d_{tan}^k = 0$, pare o algoritmo e retorne (x^k, y^k) como "solução" para o MPEC.

Se $d_{tan}^k = 0$, calcule λ^{k+1} tal que $\|\lambda^{k+1}\| \leq M$ e defina

$$s^{k+1} = z^k \quad (2.10)$$

$$\theta_k = \theta_{k,-1} \quad (2.11)$$

$$Ared_k = (1 - \theta_k)[\|C(s^k)\| - \|C(z^k)\|]. \quad (2.12)$$

e termine a iteração k .

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

Passo 4. Fase de Minimização.

Calcule $v^{k,i} \in \pi_k$ tal que

$$\|v^{k,i} - z^k\| \leq \delta_{k,i} \quad (2.13)$$

$$L(v^{k,i}, \lambda^k) \ll L(s^k, \lambda^k), \quad (2.14)$$

onde $v^{k,i} = (x^{k,i}, y^{k,i}, \mu^{k,i}, \gamma^{k,i})$.

Passo 5. Multiplicadores tentativos.

Calcule $\lambda_t^{k,i}$ tal que $\|\lambda_t^{k,i}\| \leq M$.

Passo 6. Busca do θ ótimo.

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

$$\begin{aligned} Pred_{k,i}(\theta) &= \theta[L(s^k, \lambda^k) - L(v^{k,i}, \lambda^k) - \langle C(z^k), \lambda_t^{k,i} - \lambda^k \rangle] \\ &+ (1 - \theta)[\|C(s^k)\| - \|C(z^k)\|]. \end{aligned} \quad (2.15)$$

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

$$Pred_{k,i}(\theta) \geq 0.5[\|C(s^k)\| - \|C(z^k)\|] \quad (2.16)$$

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

Passo 7. Comparação entre a redução prevista e a efetiva.

Calcule

$$Ared_{k,i} = \theta_{k,i}[L(s^k, \lambda^k) - L(v^{k,i}, \lambda_t^{k,i})] + (1 - \theta_{k,i})[\|C(s^k)\| - \|C(v^{k,i})\|]. \quad (2.17)$$

Se

$$Ared_{k,i} \geq 0.1Pred_{k,i} \quad (2.18)$$

defina $s^{k+1} = v^{k,i}$, $\lambda^{k+1} = \lambda_t^{k,i}$, $\theta_k = \theta_{k,i}$, $\delta_k = \delta_{k,i}$, $i_{ef}(k) = i$ e termine a iteração k .

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

Para facilitar a leitura vamos descrever as idéias básicas deste algoritmo.

O algoritmo é dividido em duas fases: restauração e minimização. O objetivo da fase de restauração é obter um ponto mais viável em relação às restrições de igualdade

mantendo todas as outras viáveis, já a fase de minimização contribui diminuindo o valor da função Lagrangeana no espaço definido pela linearização das restrições ($C(s) = 0$), em uma região de confiança centrada no ponto obtido na fase de restauração. Uma função de mérito é responsável pelo equilíbrio entre a viabilidade e a otimalidade, e determina a aceitação do próximo iterando da sequência gerada pelo algoritmo. Nos passos 3-7 está descrita a fase de minimização. No passo 3 uma direção d_{tan}^k é calculada. Esta direção é chamada Gradiente Projetado Aproximado (AGP), e foi usada para definir em [54] as condições de otimalidade AGP.

Um ponto x satisfaz as condições de otimalidade AGP se existe uma sequência $\{x^k\}$ que converge para x tal que $d_{tan}^k \rightarrow 0$.

Observação 1: Para que o passo 3 esteja bem definido precisamos assumir que $\{(x, y, \mu, \gamma) \in \Omega \times \Delta : g(x, y) \geq 0, \gamma \geq 0\}$ é convexo, ou seja que as funções $g(x, y)$ devem ser côncavas nas duas variáveis. Neste contexto, estamos assumindo que a viabilidade em relação a $g(x, y) \geq 0, \gamma \geq 0$ pode ser garantida. Isto na prática pode ser impossível a menos que as restrições sejam lineares.

Observação 2: No passo 2, o algoritmo interno para restaurar acha $\bar{y}$, uma solução aproximada do VIP. Uma forma de achar os multiplicadores $\bar{\mu}$ e $\bar{\gamma} \geq 0$ em Δ , é estimando-os através de quadrados mínimos.

Observação 3: O seguinte sistema KKT também está associado ao problema do segundo nível, se reescrevemos $g(x, y) \geq 0$ como $g(x, y) - z = 0$, $z \in \mathbb{R}^l$, $z \geq 0$,

$$\begin{aligned} C(x, y, z, \mu, \gamma) &= 0 \\ z &\geq 0 \quad \gamma \geq 0 \end{aligned} \tag{2.19}$$

Neste caso,

$$C(x, y, z, \mu, \gamma) = \begin{pmatrix} G(x, y) + \sum_{i=1}^q \nabla_y h_i(x, y) \mu_i - \sum_{i=1}^l \nabla_y g_i(x, y) \gamma_i \\ h(x, y) \\ g(x, y) - z \\ \gamma_1 z_1 \\ \vdots \\ \gamma_l z_l \end{pmatrix} \quad (2.20)$$

e

$$L(x, y, z, \mu, \gamma, \lambda) = f(x, y) + C(x, y, z, \mu, \gamma)^T \lambda, \quad (2.21)$$

Se aplicamos o algoritmo usando o sistema KKT (2.19), so é necessário assumir a concavidade de $g(x, y)$ na variável y , para cada x fixo.

2.3 Hipóteses sobre o problema

As hipóteses que enunciamos a seguir são as mais fracas que permitem provar resultados de convergência do algoritmo de restauração inexata apresentado em [50]. Os lemas enunciados nesta seção estão provados no artigo citado [50].

H1. X , Y e Δ são compactos.

H2. Existe $L_1 > 0$ tal que, para todo $u, v \in \Omega \times \Delta$,

$$\|C'(u) - C'(v)\| \leq L_1 \|u - v\|. \quad (2.22)$$

H3. Existe $L_2 > 0$ tal que, para todo $x, y \in X \times Y$,

$$\|\nabla f(x) - \nabla f(y)\| \leq L_2 \|x - y\|. \quad (2.23)$$

Lema 2.1. Existem $L_3, L_4 > 0$ tais que, sempre que $\|\lambda\| \leq M$, $u, v \in \Omega \times \Delta$,

$$\|C(v) - C(u) - C'(u)(v - u)\| \leq \frac{L_1}{2} \|v - u\|^2, \quad (2.24)$$

$$|f(u) - f(v) - \langle \nabla f(u), v - u \rangle| \leq \frac{L_2}{2} \|v - u\|^2, \quad (2.25)$$

$$\|\nabla L(v, \lambda) - \nabla L(u, \lambda)\| \leq L_3 \|v - u\|, \quad (2.26)$$

$$|L(v, \lambda) - L(u, \lambda) - \langle \nabla L(u, \lambda), v - u \rangle| \leq \frac{L_3}{2} \|v - u\|^2. \quad (2.27)$$

Prova: Ver Lema 3.1 [50]. ■

Lema 2.2. Se $v \in \pi_k$, então

$$\|C(v)\| \leq \|C(z^k)\| + \frac{L_1}{2} \|z^k - v\|^2. \quad (2.28)$$

Prova: Ver Lema 3.2 [50]. ■

Lema 2.3. Existem $c_2, c_3 > 0$ (independentes de k) tais que, se $v^{k,i}$ é calculado no passo 4 do algoritmo, temos que

$$L(v^{k,i}, \lambda^k) \leq L(z^k, \lambda^k) - \min\{\tau_2, c_2 \|d_{tan}^k\|^2, \tau_1 \delta_{k,i}, c_3 \|d_{tan}^k\| \delta_{k,i}\}. \quad (2.29)$$

Prova: Ver Lema 3.3 [50]. ■

2.4 Boa definição do algoritmo

Nesta seção discutimos a boa definição do Algoritmo 2.1. A primeira questão é se o passo 2 pode ser completado quando o ponto corrente não é viável. Enfatizamos que podemos escolher qualquer algoritmo globalmente convergente para resolução do $VIP(G(x^k, \cdot), D(x^k))$, no sentido de que ou pára em um número finito de iterações em um ponto estacionário do problema reformulado ou gera uma sequência tal que todo ponto de acumulação é estacionário deste problema.

Também é preciso mostrar que o passo 4 pode ser completado e que a iteração k pode ser realizada em um número finito de execuções do passo 4.

O passo 4 consiste em achar um ponto $v^{k,i}$ no espaço tangente aproximado que passa pelo ponto z^k que produz uma redução suficiente do Lagrangeano. Para isto temos liberdade para usar qualquer algoritmo globalmente convergente apropriado para resolver problemas de programação não linear nas regiões definidas pelos conjuntos π^k interceptados com uma caixa de confiança. Se as funções g_i são afins se trata de problemas com restrições lineares, já que escolhemos que a região de confiança seja uma caixa para que o problema da fase de minimização só tenha restrições lineares neste caso.

Teorema 2.1. O Algoritmo 2.1 está bem definido.

Prova: Temos que

$$\begin{aligned} & \text{Ared}_{k,i} - 0.1 \text{Pred}_{k,i} \\ &= 0.9 \text{Pred}_{k,i} + (1 - \theta_{k,i}) [\|C(z^k)\| - \|C(v^{k,i})\|] + 0.1\theta_{k,i} \langle C(z^k) - C(v^{k,i}), \lambda_t^{k,i} - \lambda^k \rangle. \end{aligned} \quad (2.30)$$

Portanto, por (2.16),

$$\text{Ared}_{k,i} - 0.1 \text{Pred}_{k,i}$$

$$\geq 0.45[\|C(s^k)\| - \|C(z^k)\|] + (1 - \theta_{k,i})[\|C(z^k)\| - \|C(v^{k,i})\|] - |0.1\theta_{k,i}\langle C(z^k) - C(v^{k,i}), \lambda_t^{k,i} - \lambda^k \rangle|.$$

Então, por (2.8),

$$\begin{aligned} & \text{Ared}_{k,i} - 0.1 \text{Pred}_{k,i} \\ & \geq 0.45(1 - r)\|C(s^k)\| - [\|C(z^k)\| - \|C(v^{k,i})\|] - |0.1\theta_{k,i}\langle C(z^k) - C(v^{k,i}), \lambda_t^{k,i} - \lambda^k \rangle|. \end{aligned} \quad (2.31)$$

Se $C(s^k) \neq 0$, o primeiro termo do lado direito de (2.31) é positivo e, pela continuidade de C , os termos restantes tendem a zero quando $\delta_{k,i} \rightarrow 0$. Portanto, existe $\delta_{k,i} > 0$ tal que $\text{Ared}_{k,i} - 0.1 \text{Pred}_{k,i} \geq 0$. Isto implica que o algoritmo está bem definido se $C(s^k) \neq 0$.

Analisamos agora o caso em que s^k é viável. Como o algoritmo não termina na iteração k , $d_{tan}^k \neq 0$. Neste caso resulta do passo 2 que $z^k = s^k$ e $C(z^k) = C(s^k) = 0$. Então, (2.16) se reduz a

$$L(z^k, \lambda^k) - L(v^{k,i}, \lambda^k) \geq 0. \quad (2.32)$$

Por (2.29), (2.32) vale independentemente de θ , portanto $\theta_{k,i} = \theta_{k,-1}$ para todo i .

Logo, por (2.18) e (2.29),

$$\text{Pred}_{k,i} = \theta_{k,-1}[L(z^k, \lambda^k) - L(v^{k,i}, \lambda^k)] \geq \theta_{k,-1} \min\{\tau_2, c_2\|d_{tan}^k\|^2, \tau_1\delta_{k,i}, c_3\|d_{tan}^k\|\delta_{k,i}\}.$$

e por (2.30),

$$\begin{aligned} & \text{Ared}_{k,i} - 0.1 \text{Pred}_{k,i} \\ & \geq 0.9\theta_{k,-1} \min\{\tau_2, c_2\|d_{tan}^k\|^2, \tau_1\delta_{k,i}, c_3\|d_{tan}^k\|\delta_{k,i}\} - [\|C(z^k)\| - \|C(v^{k,i})\|] - \\ & \quad 0.1|\langle C(z^k) - C(v^{k,i}), \lambda_t^{k,i} - \lambda^k \rangle|. \end{aligned}$$

Então,

$$\begin{aligned} & \frac{\text{Ared}_{k,i} - 0.1 \text{Pred}_{k,i}}{\delta_{k,i}} \\ & \geq 0.9\theta_{k,-1} \min\left\{\frac{\tau_2}{\delta_{k,i}}, c_2\frac{\|d_{tan}^k\|^2}{\delta_{k,i}}, \tau_1, c_3\|d_{tan}^k\|\right\} - \frac{[\|C(z^k)\| - \|C(v^{k,i})\|]}{\delta_{k,i}} - \end{aligned}$$

$$0.1 \frac{\|C(z^k) - C(v^{k,i})\| \|\lambda_t^{k,i} - \lambda^k\|}{\delta_{k,i}}.$$

O primeiro termo do lado direito da desigualdade está limitado inferiormente por um número positivo e por (2.28), os termos restantes tendem a zero. Portanto, $\text{Ared}_{k,i} - 0.1 \text{Pred}_{k,i} \geq 0$ se $\delta_{k,i}$ é suficientemente pequeno. ■

O seguinte resultado mostra que a redução efetiva $\text{Ared}_{k, \text{ief}(k)}$ obtida a cada iteração tende a zero. Uma consequência imediata é a viabilidade dos pontos de acumulação da sequência gerada pelo algoritmo. Definimos $\psi(s, \theta) \equiv \theta L(s, \lambda) + (1 - \theta) \|C(s)\|$.

Teorema 2.2. Se o Algoritmo 2.1 gera uma sequência infinita, então

$$\lim_{k \rightarrow \infty} \psi(s^k, \theta_k) - \psi(s^{k+1}, \theta_k) = 0.$$

Prova: Suponhamos que existe um conjunto infinito de índices $K_1 \subset \mathbb{N}$ e um número positivo $\rho > 0$ tal que

$$\psi(s^{k+1}, \theta_k) \leq \psi(s^k, \theta_k) - \rho$$

para todo $k \in K_1$. Denotamos $\psi_k = \psi(s^k, \theta_k)$ para todo $k \in \mathbb{N}$.

Então, para todo $k \in \mathbb{N}$ temos que

$$\begin{aligned} \psi_{k+1} &= \theta_{k+1} L(s^{k+1}, \lambda^{k+1}) + (1 - \theta_{k+1}) \|C(s^{k+1})\| \\ &= \theta_{k+1} f(s^{k+1}) + (1 - \theta_{k+1}) \|C(s^{k+1})\| - [\theta_k L(s^{k+1}, \lambda^{k+1}) + \\ &\quad (1 - \theta_k) \|C(s^{k+1})\|] + [\theta_k f(s^{k+1}) + (1 - \theta_k) \|C(s^{k+1})\|] \\ &= (\theta_{k+1} - \theta_k) L(s^{k+1}, \lambda^{k+1}) + (\theta_k - \theta_{k+1}) \|C(s^{k+1})\| + [\theta_k L(s^{k+1}, \lambda^{k+1}) + (1 - \theta_k) \|C(s^{k+1})\|] \end{aligned}$$

$$\begin{aligned}
&= (\theta_k - \theta_{k+1})(\|C(s^{k+1})\| - L(s^{k+1}, \lambda^{k+1})) + [\theta_k L(s^k, \lambda^k) + (1 - \theta_k)\|C(s^k)\|] - \rho_k \\
&= (\theta_k - \theta_{k+1})(\|C(s^{k+1})\| - L(s^{k+1}, \lambda^{k+1})) + \psi_k - \rho_k,
\end{aligned} \tag{2.33}$$

onde $\rho_k \geq 0$ para todo $k \in \mathbb{N}$ and $\rho_k \geq \rho > 0$ para todo $k \in K_1$. Pela definição de $\theta_{k,-1}$ no Algoritmo 2.1, temos

$$\theta_k - \theta_{k+1} + \omega_k \geq 0. \tag{2.34}$$

para todo $k \in \mathbb{N}$. Pela compacidade de $\Omega \times \Delta$, existe um $c > 0$ tal que

$$|\|C(s^k)\| - L(s^k, \lambda^k)| \leq c$$

para todo $k \in \mathbb{N}$. Então, por (2.33) e (2.34), temos que

$$\begin{aligned}
\psi_{j+1} &= (\theta_j - \theta_{j+1} + \omega_j)(\|C(s^{j+1})\| - L(s^{j+1}, \lambda^{j+1})) + \psi_j - \rho_j - \omega_j(\|C(s^{j+1})\| - L(s^{j+1}, \lambda^{j+1})) \\
&\leq (\theta_j - \theta_{j+1} + \omega_j)c + \psi_j - \rho_j + \omega_j c = (\theta_j - \theta_{j+1})c + \psi_j - \rho_j + 2\omega_j c
\end{aligned}$$

para $j = 0, 1, \dots, k-1$. Somando estas k desigualdades, obtemos

$$\psi_k \leq \psi_0 + (\theta_0 - \theta_k)c + \sum_{j=0}^{k-1} 2c\omega_j - \sum_{j=0}^{k-1} \rho_j \leq \psi_0 + 2c + \sum_{j=0}^{k-1} 2c\omega_j - \sum_{j=0}^{k-1} \rho_j \tag{2.35}$$

para todo $k \geq 1$. Como a série $\sum_{j=0}^{\infty} \omega_j$ é convergente, e ρ_k está limitado inferiormente por um número positivo, para $k \in K_1$, (2.35) implica que ψ_k é ilimitado inferiormente, contradizendo a compacidade de $\Omega \times \Delta$. ■

Uma consequência do teorema anterior é que, se o Algoritmo 2.1 gera uma sequência infinita, $\lim_{k \rightarrow \infty} \|C(s^k)\| = 0$, significando que pontos arbitrariamente próximos à região viável do problema reformulado são gerados.

Teorema 2.3. Se o Algoritmo 2.1 não pára em finitos passos , então

$$\lim_{k \rightarrow \infty} \|C(s^k)\| = 0.$$

(Em particular, todo ponto limite s de $\{s^k\}$ satisfaz $C(s) = 0$.)

Prova: Por (2.8), (2.18) e (2.16) temos que

$$\begin{aligned}\|C(s^k)\| &\leq \frac{\|C(s^k)\| - \|C(z^k)\|}{1-r} \leq \frac{2}{1-r} \text{Pred}_{k, i_{ef}(k)} \leq \frac{20}{1-r} \text{Ared}_{k, i_{ef}(k)} \\ &= \frac{20}{1-r} [\psi(s^k, \theta_k) - \psi(s^{k+1}, \theta_k)].\end{aligned}$$

Portanto, o resultado desejado segue do Teorema 2.1. ■

2.5 Convergência do Algoritmo

Nesta seção daremos condições para que uma sequência gerada pelo algoritmo admita uma subsequência tal que $\|d_{tan}^k\| \rightarrow 0$. Na prática, isto significa que dadas tolerâncias $\varepsilon_v, \varepsilon_{opt} > 0$, para algum iterando s^k teremos $\|C(s^k)\| \leq \varepsilon_v$ e $\|d_{tan}^k\| \leq \varepsilon_{opt}$.

Hipótese H4. Existe $\beta > 0$ tal que

$$\|z^k - s^k\| \leq \beta \|C(s^k)\| \quad \forall k \in \mathbb{N}. \quad (2.36)$$

Assumiremos na Hipótese C abaixo que não existe subsequência de $\{d_{tan}^k\}$ que tende a zero, e veremos que esta suposição nos leva a uma contradição.

Hipótese C. Existe $\varepsilon > 0$ e $k_0 \in \mathbb{N}$ tal que $|d_{tan}^k| \geq \varepsilon$ para todo $k \geq k_0$.

Lema 2.4. Suponhamos que a Hipótese C é verdadeira. Então, existe $c_4, c_5 > 0$ tal que $L(z^k, \lambda^k) - L(v^{k,i}, \lambda^k) \geq \min\{c_4, c_5 \delta_{k,i}\}$ para todo $k \geq k_0$, $i = 0, 1, \dots, i_{ef}(k)$.

Prova: Segue do Lema 2.3 e da Hipótese C. ■

Lema 2.5. Suponhamos que a Hipótese C é verdadeira. Então, existe $\alpha, \varepsilon_1 > 0$ tal que sempre que $k \geq k_0$ e $|C(s^k)| \leq \min\{\varepsilon_1, \alpha\delta_{k,i}\}$ temos que $\theta_{k,i} = \theta_{k,i-1}$.

Prova: Por (2.8), (2.36), (3.3), Lema 2.4, a compacidade de $X, Y\Delta$, a continuidade da C e f e pela limitação dos λ^k , existe $c, c', c'' > 0$ tal que

$$\begin{aligned}
& \text{Pred}_{k,i}(1) - \frac{1}{2}[|C(s^k)| - |C(z^k)|] \\
& \geq [L(s^k, \lambda^k) - L(v^{k,i}, \lambda^k) - \langle C(z^k), \lambda_{trial}^{k,i} - \lambda^k \rangle - \frac{1}{2}|C(s^k)|] \geq \\
& \geq [L(z^k, \lambda^k) - L(v^{k,i}, \lambda^k)] - |L(s^k, \lambda^k) - L(z^k, \lambda^k)| - c'|C(s^k)| - \frac{1}{2}|C(s^k)| \\
& \geq \min\{c_4, c_5\delta_{k,i}\} - c'|z^k - s^k| - (c'' + \frac{1}{2})|C(s^k)| \\
& \geq \min\{c_4, c_5\delta_{k,i}\} - c'\beta|C(s^k)| - (c'' + \frac{1}{2})|C(s^k)| \\
& = \min\{c_4, c_5\delta_{k,i}\} - c|C(s^k)|,
\end{aligned}$$

with $c = c'\beta + c'' + \frac{1}{2}$. Portanto,

$$\frac{\text{Pred}_{k,i}(1) - \frac{1}{2}[|C(s^k)| - |C(z^k)|]}{c} \geq \min\{\frac{c_4}{c}, \frac{c_5}{c}\delta_{k,i}\} - |C(s^k)|.$$

Assim, se $\varepsilon_1 = c_4/c$, $\alpha = c_5/c$ e $|C(s^k)| \leq \min\{\varepsilon_1, \alpha\delta_{k,i}\}$, temos que $\text{Pred}_{k,i}(1) - \frac{1}{2}[|C(s^k)| - |C(z^k)|] \geq 0$. Isto significa que (2.18) é verdadeira para $\theta = 1$. Como é verdadeiro para $\theta = 0$, vale para todo $\theta \in [0, 1]$. Isto implica que $\theta_{k,i} = \theta_{k,i-1}$, como desejamos provar. ■

Lema 2.6. Suponhamos que a Hipótese C é verdadeira. Então existe $\bar{\theta} > 0$ tal que $\theta_k \geq \bar{\theta}$ para todo $k \in \mathbb{N}$.

Prova: Observe que

$$\text{Ared}_{k,i} - \text{Pred}_{k,i} = \theta_{k,i} \langle C(s^k) - C(v^{k,i}), \lambda_{trial}^{k,i} - \lambda^k \rangle + (1 - \theta_{k,i}) [|C(z^k)| - |C(v^{k,i})|].$$

Portanto, por (2.22) and (2.28), existe $c > 0$ tal que

$$\text{Ared}_{k,i} \geq \text{Pred}_{k,i} - c\delta_{k,i}^2$$

para todo $k \in \mathbb{N}$, $i = 0, 1, \dots, i_{ef}(k)$. A prova deste Lema segue como no Lema 4.3 de [52] trocando c_1 por c e $C(\cdot)^+$ por $C(\cdot)$.

Teorema 2.4. A Hipótese C é falsa

Prova: Ver Teorema 4.4 [52]. ■

Teorema 2.5. Suponhamos que as hipóteses H1, H2, H3 são satisfeitas e que a sequência gerada pelo Algoritmo 2.1 verifica a hipótese H4, então

1. $\|C(s^k)\| \rightarrow 0$.
2. Todo ponto limite de $\{s^k\}$ é viável.
3. Se s^* é um ponto limite de $\{s^k\}$ e $\lim_{k \in K_1} s^k = s^*$, então $\lim_{k \in K_1} z^k = s^*$.
4. Existe um conjunto infinito $K_2 \subset \mathbb{N}$ tal que $\lim_{k \in K_2} d_{tan}^k = 0$.
5. Existe um conjunto infinito $K_3 \subset \mathbb{N}$ e $s^* \in \Omega \times \Delta$ tal que

$$\lim_{k \in K_3} s^k = \lim_{k \in K_3} z^k = s^*, \quad C(s^*) = 0 \quad \text{e} \quad \lim_{k \in K_3} d_{tan}^k = 0.$$

Prova: Os dois primeiros itens seguem do Teorema 2.3. O terceiro e quarto itens seguem do Teorema 2.4. Finalmente para o último item é suficiente tomar uma subsequência adequada de $\{s^k\}$. ■

Observação: Se na definição do conjunto $D(x)$ no problema original não há restrições de desigualdade, podemos dizer que se s^* é um ponto regular, então ele satisfaz as condições de otimalidade Karush-Kuhn-Tucker do problema reformulado. Neste caso o sistema KKT não apresenta restrições de complementaridade.

Na próxima seção daremos condições suficientes para que 2.36 seja válida.

2.5.1 Condições suficientes para convergência a pontos AGP

As seguintes hipóteses sobre o problema (2.1) garantem a existência de um limitante β que verifica (2.36):

- A1.** Para cada $(x, y(x))$ com $x \in X$ e $y(x)$ solução do $VIP(G(x, \cdot), D(x))$, temos que $(x, y(x))$ é regular, no sentido de que o conjunto das restrições ativas em $(x, y(x))$ são linearmente independentes.
- A2.** $(x, y(x), \mu(x), \gamma(x))$ é a única solução do sistema (2.4). Ou seja existe uma função $w : X \rightarrow Y \times \Delta$ tal que $w(x) = (y(x), \mu(x), \gamma(x))^T$.
- A3.** O operador $[\nabla_y G(x, y(x)) - \sum_{i=1}^I \nabla_y^2 g_i(x, y(x)) \gamma(x)_i]$ é estritamente convexo em $Nu(h'_y(x, y(x)) \cap Nu((g_I)'_y(x, y(x))))$, onde I é o conjunto de índices das restrições ativas em $(x, y(x))$.
- A4.** Para todo $(x, y(x), \mu(x), \gamma(x))$ solução do sistema (2.4) temos,

$$g(x, y(x))_i + \gamma(x)_i > 0 \quad \forall i \in \{1, \dots, m\}. \quad (2.37)$$

Com estas suposições sobre o problema, podemos afirmar que,

- (i) A matriz $C'_{y,\mu,\gamma}(x, y(x), \mu(x), \gamma(x))$ é não singular para todo $x \in X$.
- (ii) A função $w(x) = (y(x), \mu(x), \gamma(x))^T$ é continuamente diferenciável em X . Resulta que $X \times w(X)$ é compacto.
- iii) Existe β independente de x que satisfaz $\|s - (x, y(x), \mu(x), \gamma(x))\| \leq \beta \|C(s)\|$ para $s \in \Upsilon$, onde Υ é um conjunto aberto tal que $X \times w(X) \subset \Upsilon$ e β depende de limitantes de C' e da matriz inversa de $C'_{y,\mu,\gamma}(x, y(x), \mu(x), \gamma(x))$.

As demonstrações das afirmações (i) e (iii) encontram-se no Apêndice desta tese e foram baseadas no Teorema 3.2.2 e Lema 3.2.4 de [33], já a afirmação (ii) é uma consequência direta da aplicação do teorema da função implícita enunciado no Apêndice.

Observação : A hipótese **A3** significa que $w^T \Gamma(x) w \neq 0 \forall w \neq 0 \in Nu(h'_y(x, y(x)) \cap Nu((g_I)'_y(x, y(x))))$, onde $\Gamma(x) = [\nabla_y G(x, y(x)) + \sum_{i=1}^l \nabla_y^2 g_i(x, y(x)) \gamma(x)_i]$, portanto no caso particular em que as funções que definem o problema MPEC são afins, o resultado também vale porque o conjunto de vetores w que satisfaz A3 é vazio.

O valor β obtido vale para $s \in \Upsilon$. Precisamos provar que existe um valor limitante β para todo $\Omega \times \Delta$. Este resultado foi provado em [18] e o reproduzimos aqui. No próximo lema, mostramos que existe um limitante global para o erro.

O resultado que segue é essencial para provar a boa definição do algoritmo que vamos propor para resolver problemas de dois níveis com as hipóteses dadas.

Teorema 2.6. Seja $x \in X$ e $(y(x), \mu(x), \gamma(x)) \in Y \times \Delta$. Se $\{(x, y^k, \mu^k, \gamma^k)\}$ é uma seqüência convergente ao ponto $(x, y(x), \mu(x), \gamma(x))$ tal que $C(x, y(x), \mu(x), \gamma(x)) = 0$, então existe β que satisfaz a condição (2.36).

Prova: Se $C(x, y, \mu, \gamma) \neq 0$, existe N tal que

$$\forall n > N \Rightarrow \|C(x, y^n, \mu^n, \gamma^n) - C(x, y(x), \mu(x), \gamma(x))\| < r\|C(x, y, \mu, \gamma)\|$$

como $C(x, y(x), \mu(x), \gamma(x)) = 0$ temos que:

$$\forall n > N \Rightarrow \|C(x, y^n, \mu^n, \gamma^n)\| < r\|C(x, y, \mu, \gamma)\|$$

Temos que para cada $x \in X$ existe $\beta > 0$, independente de x , tal que,

$$\|(x, y, \mu, \gamma) - (x, y(x), \mu(x), \gamma(x))\| \leq \beta\|C(x, y, \mu, \gamma)\|$$

Utilizando a desigualdade triangular, temos que:

$$\begin{aligned} & \|(x, y, \mu, \gamma) - (x, y^n, \mu^n, \gamma^n)\| \leq \\ & \|(x, y, \mu, \gamma) - (y(x), \mu(x), \gamma(x))\| + \|(y(x), \mu(x), \gamma(x)) - (y^n, \mu^n, \gamma^n)\| \\ & \leq \beta\|C(x, y, \mu, \gamma)\| + r\|C(x, y, \mu, \gamma)\| \end{aligned}$$

Se renomeamos $(\beta + r\beta)$, obtemos o resultado desejado. ■

Observação: É interessante notar que os resultados obtidos acima não utilizam o fato das funções h_i serem afins. Esta observação permite concluir que o algoritmo também pode ser aplicado a um problema de otimização cujas restrições sejam um sistema KKT

geral, ou seja um problema com restrições de complementaridade (MPCC).

Nos próximos capítulos resolvemos problemas teste e uma aplicação, onde o problema de segundo nível é sempre um $VIP(G(x, \cdot), D(x))$, portanto a região viável $D(x)$ é convexa.

Capítulo 3

Experimentos Numéricos e Aplicações

3.1 Experimentos Numéricos

O algoritmo de Restauração Inexata para MPEC proposto neste trabalho foi implementado utilizando um computador Pentium 4, 2.0 GHz e 512MB e a versão 6.1 do Matlab.

Conforme mencionado no capítulo 2 este algoritmo é composto de duas fases: Restauração e Minimização.

A Fase de Restauração é dividida em dois passos, o primeiro encontra $\bar{y}$ solução aproximada do $VIP(G(x^k, .), D(x^k))$, utilizando o algoritmo de projeções proposto por Solodov e Svaiter [66] discutido no capítulo 1, o qual foi implementado em Matlab. No segundo passo estamos interessados em atualizar os multiplicadores (μ, γ) de tal forma que satisfaçam a condição de restauração inexata. Nesta fase utilizamos uma estratégia de Quadrados Mínimos, ou seja, determinamos $(\bar{\mu}, \bar{\gamma})$ solução inexata do problema

$$\underset{\mu, \gamma \in \Delta}{\text{Minimizar}} \|C(x^k, \bar{y}, \mu, \gamma)\|^2$$

que satisfaça à condição

$$\|C(x^k, \bar{y}, \bar{\mu}, \bar{\gamma})\| \leq r \|C(x^k, y^k, \mu^k, \gamma^k)\|, \quad (3.1)$$

Portanto este é o critério de parada utilizado no método para resolver (3.1).

Na fase de minimização o objetivo é diminuir o valor da função Lagrangeana resolvendo aproximadamente o problema

$$\begin{aligned} &\underset{x, y, \mu, \gamma}{\text{Minimizar}} L(x, y, \mu, \gamma) \quad s.a \\ &\|(x, y, \mu, \gamma) - (x^k, \bar{y}, \bar{\mu}, \bar{\gamma})\| \leq \delta_{k,i} \quad \text{e } (x, y, \mu, \gamma) \in \pi_k, \end{aligned} \quad (3.2)$$

onde $\delta_{k,i}$ é o raio da região de confiança. A solução aproximada de (3.2) aceita é $(x_{k+1}, y_{k+1}, \mu_{k+1}, \gamma_{k+1})$ que satisfaz o critério de decréscimo suficiente do passo 7 do Algoritmo de Restauração Inexata para MPEC.

Os problemas de otimização (3.1) e (3.2) foram resolvidos utilizando as rotinas de otimização **fmincon** e **quadproj** internas do Matlab.

3.1.1 Algoritmos Implementados

Implementamos o Algoritmo de Restauração Inexata proposto no capítulo 2.

Também testamos uma variante do Algoritmo 2.1, baseada no trabalho de Martínez e Pilotta [52]. Este algoritmo utiliza a função objetivo do primeiro nível como função de mérito em lugar do Lagrangeano. Vamos nos referir a este algoritmo como Algoritmo 2.1(b).

Seguem os passos do Algoritmo 2.1(b) que diferem do Algoritmo 2.1

Passo 3: Direção de Cauchy.

Calcule

$$d_{tan}^k = P_k[z^k - \eta \nabla f(x^k, \bar{y})] - z^k$$

Passo 4: Fase de Minimização.

Calcule $v^{k,i} \in \pi_k$ tal que

$$\begin{aligned} \|v^{k,i} - z^k\| &\leq \delta_{k,i} \quad e \\ f(x^{k,i}, y^{k,i}) &<< f(x^k, y^k). \end{aligned}$$

Passo 6: Busca do θ ótimo.

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

$$\begin{aligned} Pred_{k,i}(\theta) &= \theta[f(x^k, y^k) - f(x^{k,i}, y^{k,i})] \\ &+ (1 - \theta)[\|C(s^k)\| - \|C(z^k)\|]. \end{aligned}$$

Passo 7: A definição da redução efetiva é:

$$Ared_{k,i} = \theta_{k,i}[f(x^k, y^k) - f(x^{k,i}, y^{k,i})] + (1 - \theta_{k,i})[\|C(s^k)\| - \|C(v^{k,i})\|].$$

3.1.2 Problemas Teste

Para testar e validar os Algoritmos 2.1 e 2.1(b) utilizamos algumas classes de problemas teste da literatura. A primeira classe consiste em problemas clássicos abordados em [28, 30, 72, 71], a segunda consiste em problemas de programação quadrática em dois níveis propostos por Vicente e Calamai em [14, 15]. Para finalizar utilizamos uma classe de problemas de programação em dois níveis relacionados com problemas de transporte [23].

Problemas Clássicos

Estes problemas foram propostos em [28, 30, 72, 71]. Para a resolução utilizamos os seguintes critérios, parâmetros de entrada e notações.

Critério de Parada dos Algoritmos 2.1 e 2.1(b) :

$$\|d_{tan}^k\| \leq 10^{-4}, \|C(s^k)\| \leq 10^{-4};$$

Número máximo de iterações externas = 250 ;

Constante de Restauração $r = 0.9$;

Critério de Parada do Algoritmo de Projeções : $\|R(y^k)\| \leq 10^{-4}$;

n: dimensão da variável (x) ;

m: dimensão da variável (y);

q : números de restrições (diferentes de caixa) ;

It_{RI} : iterações do Algoritmo de Restauração Inexata;

It_{Min} : Contador de iterações da Fase de Minimização;

Pontos Iniciais : x_0 : ponto inicial x , y_0 : ponto inicial y , z_0 e w_0 : pontos iniciais dos multiplicadores ref. às rest. de caixa, γ_0 : ponto inicial γ ;

$N.C$: Não convergiu em um número máximo de iterações;

f : valor da função objetivo do primeiro nível no ponto obtido pelo algoritmo de Restauração Inexata;

f^* : valor da função objetivo do primeiro nível no ponto obtido em [28, 30, 72, 71].

A variável y contempla as variáveis de folga do problema. A partir de agora vamos descrever com detalhes os problemas resolvidos.

Descrição dos Problemas:

Teste 1) [71].

$$\underset{x,y}{\text{Minimizar}} \quad -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3$$

$$\text{s.a } x \in X = \{x \in \mathbb{R}^2 : x \geq 0\}$$

$$y = \underset{y \in D(x)}{\text{argmin}} \quad x_1 + 2x_2 + y_1 + y_2 + 2y_3$$

onde $D(x) = \{y \in \mathbb{R}^3 : y_2 + y_3 - y_1 \leq 1, \quad 2x_1 - y_1 + 2y_2 - 0.5y_3 \leq 1, \quad 2x_2 + 2y_1 - y_2 - 0.5y_3 \leq 1 \text{ e } y \geq 0\}$.

$$n = 2, \quad m = 6 \text{ e } q = 3;$$

$$x_0 = (0, 0.8)^t;$$

$$y_0 = z_0 = w_0 = (0, 0, 0, 0, 0, 0)^t;$$

$$\gamma_0 = (0, 0, 0)^t.$$

Testes 2-6) [71].

$$\underset{x,y}{\text{Minimizar}} \quad rx^T x - 3y_1 - 4y_2 + 0.5y^t y$$

$$\text{s.a } x \in \mathbb{R}^2$$

$$y = \underset{y \in D(x)}{\text{argmin}} \quad 0.5y^T H y - b(x)^T y$$

$$D(x) = \{y \in \mathbb{R}^2 : -0.333y_1 + y_2 \leq 2 \text{ e } y_1 - 0.333y_2 \leq 2\}$$

onde

$$H_1 = \begin{pmatrix} 1 & -2 \\ -2 & 5 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 1 & 3 \\ 3 & 10 \end{pmatrix} \quad \text{e}$$

$$H_3 = \begin{pmatrix} -1 & 2 \\ 3 & -3 \end{pmatrix}.$$

$$n = 2, \quad m = 4 \text{ e } q = 2;$$

$$x_0 = (0, 0)^t;$$

$$y_0 = z_0 = w_0 = (0, 0, 0, 0)^t;$$

$$\gamma_0 = (0, 0)^t.$$

$$2- \quad r = 0.1, \quad H = H_1 \text{ e } b(x) = x.$$

$$3- \quad r = 1, \quad H = H_1 \text{ e } b(x) = x.$$

$$4- \quad r = 0, \quad H = H_2 \text{ e } b(x) = x.$$

$$5- \quad r = 0.1, \quad H = H_2 \text{ e } b(x) = x.$$

$$6- \quad r = 0.1, \quad H = H_3 \text{ e } b(x) = x.$$

Teste 7) [71].

$$\underset{x,y}{\text{Minimizar}} \quad 100x + 1000y_1 \text{ s.a}$$

$$x \in X = \{x \in \mathbb{R} : 0 \leq x \leq 1\}$$

$$y = \underset{y \in D(x)}{\text{argmin}} \quad y_1 + 2y_2$$

$$\text{onde } D(x) = \{y \in \mathbb{R}^2 : x + y_1 - y_2 \leq 1, \quad y_1 + y_2 \leq 1,$$

$$2x_2 + 2y_1 - y_2 - 0.5y_3 \leq 1 \text{ e } y \geq 0\}.$$

$$n = 1, \quad m = 4 \text{ e } q = 2;$$

$$x_0 = 1;$$

$$y_0 = z_0 = w_0 = (0, 0, 0, 0)^t;$$

$$\gamma_0 = (0, 0)^t.$$

Teste 8) [71].

$$\underset{x,y}{\text{Minimizar}} (x-1)^2 + (y-1)^2$$

$$\text{s.a } x \in \mathbb{R}$$

$$y = \underset{y \in D(x)}{\text{argmin}} 0.5y^2 + 500y - 50xy$$

$$n = 1, m = 1 \text{ e } q = 0;$$

$$x_0 = 0;$$

$$y_0 = z_0 = w_0 = 0.$$

Teste 9) [71].

$$\underset{x,y}{\text{Minimizar}} (x-1)^2 + 2y_1 - 2x$$

$$\text{s.a } x \in X = \{x \in \mathbb{R} : x \geq 0\}$$

$$y = \underset{y \in D(x)}{\text{argmin}} (2y_1 - 4)^2 + (2y_2 - 1)^2 + xy_1$$

$$\text{onde } D(x) = \{y \in \mathbb{R}^2 : 4x + 5y_1 + 4y_2 \leq 12, 4x - 4y_1 + 5y_2 \leq 12, \\ -4x - 5y_1 + 4y_2 \leq -4, -4x + 5y_2 + 4y_1 \leq 4 \text{ e } y \geq 0\}.$$

$$n = 2, m = 6 \text{ e } q = 4;$$

$$x_0 = (0, 0)^t;$$

$$y_0 = z_0 = w_0 = (0, 0, 0, 0, 0, 0)^t;$$

$$\gamma_0 = (0, 0, 0, 0)^t.$$

Teste 10) [28].

$$\underset{x,y}{\text{Minimizar}} x_1^2 - 2x_1 + x_2^2 - 2x_2 + y_1^2 + y_2^2$$

$$\text{s.a } x \in X = \{x \in \mathbb{R} : x \geq 0\}$$

$$y = \underset{y \in D(x)}{\text{argmin}} (y_1 - x_1)^2 + (x_2 - y_2)^2$$

$$\text{onde } D(x) = \{y \in \mathbb{R}^2 : 0.5 \leq y \leq 1.5\}.$$

$$n = 2, m = 2 \text{ e } q = 0;$$

$$x_0 = (0, 0)^t;$$

$$y_0 = z_0 = w_0 = (0, 0)^t.$$

Teste 11) [28].

$$\text{Minimizar}_{x,y} x_1^2 - 3x_1 + x_2^2 - 3x_2 + y_1^2 + y_2^2$$

$$\text{s.a } x \in X = \{x \in \mathbb{R} : x \geq 0\}$$

$$y = \operatorname{argmin}_{y \in D(x)} (y_1 - x_1)^2 + (x_2 - y_2)^2$$

$$\text{onde } D(x) = \{y \in \mathbb{R}^2 : 0.5 \leq y \leq 1.5\}.$$

$$n = 2, \quad m = 2 \text{ e } q = 0;$$

$$x_0 = (0, 0)^t;$$

$$y_0 = z_0 = w_0 = (0, 0)^t.$$

Teste 12) [30, 72].

$$\text{Minimizar}_{x,y} \frac{1}{2}[x^2 + xy] - 95x$$

$$\text{s.a } x \in X = \{x : 0 \leq x \leq 200\}$$

$$y = \operatorname{argmin}_{y \in D(x)} \frac{1}{2}[y^2 + xy] - 100y$$

$$\text{onde } D(x) = \{y \in \mathbb{R} : y \geq 0\}.$$

$$n = 1, \quad m = 1 \text{ e } q = 0;$$

$$x_0 = 0;$$

$$y_0 = z_0 = w_0 = 0.$$

Teste 13) [30, 72].

$$\text{Minimizar}_{x,y} -4xy + 3y + 2x + 1$$

$$\text{s.a } x \in X = \{x : x \geq 0\}$$

$$y = \operatorname{argmin}_{y \in D(x)} -y + 4xy$$

$$\text{onde } D(x) = \{y \in \mathbb{R}^2 : 0 \leq y \leq \frac{1}{2}\}.$$

$$n = 1, m = 1 \text{ e } q = 0;$$

$$x_0 = 0;$$

$$y_0 = z_0 = w_0 = 0.$$

Teste 14) [30, 72].

$$\underset{x,y}{\text{Minimizar}} \frac{1}{2}[(y_1 - 2)^2 + (y_2 - 2)^2]$$

$$\text{s.a } x \in X = \{x : x \geq 0\}$$

$$y = \underset{y \in D(x)}{\text{argmin}} y_1^2 + y_2$$

$$\text{onde } D(x) = \{y \in \mathbb{R}^2 : -x + y_1 + y_2 = 0 \text{ e } y \geq 0\}.$$

$$n = 1, m = 2 \text{ e } q = 1;$$

$$x_0 = 2;$$

$$y_0 = z_0 = w_0 = (0, 0)^t;$$

$$\gamma_0 = 0.$$

Teste 15) [30, 72].

$$\underset{x,y}{\text{Minimizar}} (x - 5)^2 + (2y + 1)^2$$

$$\text{s.a } x \in X = \{x : x \geq 0\}$$

$$y = \underset{y \in D(x)}{\text{argmin}} (y - 1)^2 - 15xy$$

$$\text{onde } D(x) = \{y \in \mathbb{R}^2 : -3x + y \leq -3, x - 0.5y \leq 4 \text{ e } x + y \leq 7\}.$$

$$n = 1, m = 4 \text{ e } q = 3;$$

$$x_0 = 3;$$

$$y_0 = (1, 1, 1, 1)$$

$$z_0 = w_0 = (0, 0, 0, 0)^t;$$

$$\gamma_0 = (0, 0, 0).$$

Teste 16) [30, 72].

$$\underset{x,y}{\text{Minimizar}} (x-3)^2 + (y-2)^2$$

$$\text{s.a } x \in X = \{x : x \geq 0\}$$

$$y = \underset{y \in D(x)}{\text{argmin}} (y-5)^2$$

$$\text{onde } D(x) = \{y \in \mathbb{R}^2 : -2x + y \leq 1, x - 2y \leq 2 \text{ e } x + 2y \leq 14\}.$$

$$n = 1, m = 4 \text{ e } q = 3;$$

$$x_0 = 1;$$

$$y_0 = z_0 = w_0 = (0, 0, 0, 0)^t;$$

$$\gamma_0 = (0, 0, 0).$$

Teste 17) [30, 72].

$$\underset{x,y}{\text{Minimizar}} (x_1 - y_1)^2 + (x_2 - y_2)^2$$

$$\text{s.a } x \in X = \{x : 0 \leq x \leq 10\}$$

$$\langle G(x, y), y - z \rangle \leq 0 \quad \forall z \in D(x)$$

$$\text{onde } D(x) = \{y \in \mathbb{R}^2 : x_2 + y_1 \leq 15 \text{ e } y_2 + x_1 \leq 15\} \text{ e}$$

$$G(x, y) = (2y_1 + \frac{8}{3}y_2 - 34, 1.25y_1 + 2y_2 - 14.25)^t.$$

$$n = 2, m = 4 \text{ e } q = 2;$$

$$x_0 = 0;$$

$$y_0 = z_0 = w_0 = (0, 0, 0, 0)^t;$$

$$\gamma_0 = (0, 0).$$

Teste 18) [30, 72].

$$\underset{x,y}{\text{Minimizar}} 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60 + R[\max(0, x_1 + x_2 + y_1 - 2y_2 - 40)]^2$$

$$\text{s.a } x \in X = \{x : 0 \leq x \leq 50\}$$

$$\langle G(x, y), y - z \rangle \leq 0 \quad \forall z \in D(x)$$

onde $D(x) = \{y \in \mathbb{R}^2 : 2y_1 - x_1 + 10 \leq 0, 2y_2 - x_2 + 10 \leq 0 \text{ e } -10 \leq y \leq 20\}$ e

$$G(x, y) = (2y_1 - 2x_1 + 40, 2y_2 - 2x_2 + 40)^t.$$

$$n = 2, m = 4 \text{ e } q = 2;$$

$$x_0 = (20, 35)^t;$$

$$y_0 = z_0 = w_0 = (0, 0, 0, 0)^t;$$

$$\gamma_0 = (0, 0).$$

Observamos na Tabela 3.1 que o Algoritmo 2.1 obteve na maioria dos casos a mesma solução obtida em [28, 30, 72, 71], a menos dos problemas 9, 5 e 6 onde o número de iterações máximo foi atingido e foram encontrados outros pontos estacionários. O mesmo aconteceu com o Algoritmo 2.1(b) segundo a tabela 3.2, mas é interessante observar que neste caso a viabilidade em relação a C é pior que no algoritmo 2.1 devido à utilização da função objetivo como função de mérito, resultado esperado quando comparado com o desempenho da função Lagrangeana.

Programação Quadrática em Dois Níveis

Consideremos o seguinte problema de programação em dois níveis:

$$\begin{aligned} & \underset{x, y}{\text{Minimizar}} \quad Q(x, y) \\ & \text{s.a } x \geq 0 \quad (BLG) \end{aligned}$$

$$y = \underset{y \in D(x)}{\text{argmin}} \quad q(x, y)$$

onde Q e q são contínuas, diferenciáveis, quadráticas e convexas, $D(x) = \{y \in \mathbb{R}^m : A_x x + A_y y = b(\rho), y \geq 0\}$, $\rho \in \mathbb{R}^l$, $l = \min\{m, n\}$ e $\rho_i \geq 1$.

Definimos:

$$M_1 = \{k \in \{1, 2, \dots, l\} : \rho_k = 1\}, m_1 = \#M_1;$$

$$M_2 = \{k \in \{1, 2, \dots, l\} : 1 < \rho_k < 2\}, m_2 = \#M_2;$$

$$M_3 = \{k \in \{1, 2, \dots, l\} : \rho_k = 2\}, m_3 = \#M_3;$$

$$M_4 = \{k \in \{1, 2, \dots, l\} : \rho_k > 2\}, m_4 = \#M_4 .$$

Esta técnica foi proposta em [14, 15] e consiste em gerar problemas de programação em dois níveis quadráticos e densos. Uma característica importante destes problemas é a complexidade da resolução em função do número de minimizadores locais não globais que não dependem da estrutura de Q e q . Ver [14, 15]. Mas por outro lado, estes problemas possuem valores de mínimos globais conhecidos e a quantidade de minimizadores locais não globais pode ser calculado. Mais precisamente temos os seguintes resultados.

R1- O Problema BLG possui 2^{m_3} soluções globais;

R2- O Problema BLG possui $2^{m_2+m_3+m_4} - 2^{m_3}$ soluções locais não globais;

R3- $Q(x^*, y^*) = \sum_{k \in M_2} (\frac{(\rho_k - 1)}{2})^2 + \frac{(m_3 + m_4)}{4}$, onde (x^*, y^*) é uma solução global de BLG.

Critério de Parada do Algoritmo de Restauração Inexata, parâmetros de entrada e

Notações :

$$\|d_{tan}^k\| \leq 10^{-4}, \|C(s^k)\| \leq 10^{-4};$$

Número máximo de iterações externas: $k=250$;

Constante de restauração: $r = 0.9$;

Critério de Parada do Algoritmo de Projeções : $\|R(y^k)\| \leq 10^{-4}$;

It_{RI} : iterações do Algoritmo de Restauração Inexata;

It_{Min} : Contador de iterações da Fase de Minimização;

n_x : dimensão de x ;

m_y : dimensão de y ;

$(\bar{x}, \bar{y})$: Solução obtida pelo Algoritmo de Restauração Inexata;

(x^*, y^*) : Solução ótima do problema BLG;

$N.S.G$: Número de soluções globais do BLG;

$N.S.L$: Número de minimizadores locais não globais BLG.

As tabelas 3.3 e 3.4 mostram que o algoritmo 2.1 e 2.1(b) foram capazes de obter um minimizador global em 60% dos testes sendo que no restante pelo menos um ponto estacionário foi encontrado em todos os problemas.

3.1.3 Problemas de Estrutura de Redes

O Problema de Estrutura de Redes [23, 49, 55] pertence a uma classe de problemas que consistem em alterar a infraestrutura de transportes de uma rede com o objetivo de melhorar a qualidade de vida dos usuários da rede e além disso minimizar os custos da estrutura e manutenção.

Nos problemas que resolvemos nesta seção as demandas são fixas e possuem uma origem e um destino, assim, consideramos a seguinte formulação:

$$\text{Minimizar}_{z,x} \sum_{i \in A} [x_i t_i(x_i, z_i) + g_i(z_i)]$$

$$\text{s.a } z_i > -1$$

$$\text{Minimizar}_x \sum_{i \in A} \int_0^{x_i} t_i(u, z_i) du$$

$$x_i \geq 0 \quad i \in A$$

$$\sum_{j \in P} f_j = d$$

$$x_i = \sum_{j \in P} \delta_{i,j} f_{j,i}$$

Para ajudar na formulação do problema definimos

P : conjunto de rotas da rede;

A : conjunto dos índices dos arcos pertencentes à rede;

x_i : fluxo total sobre o arco $i \in \{1, 2, \dots, n\}$;

t_i : custo em tempo de uma unidade de fluxo v_i sobre o arco i ;

$g_i = l_i z_i$: capacidade sobre o arco i ;

f_i : fluxo total sobre a rota $i \in P$;

z_i : capacidade sobre o arco i ;

l_i : parametros fixos;

d : demanda da origem até o destino.

Testamos os problemas NPD1 e NPD2 propostos em [23]

Cr terios de parada e par metros de entrada:

$$\|d_{tan}^k\| \leq 10^{-4}, \|C(s^k)\| \leq 10^{-4};$$

Número máximo de iterações externas (k=250);

Constante de Restauração $r = 0.9$;

Critério de Parada do Algoritmo de Projeções : $\|R(y^k)\| \leq 10^{-4}$;

It_{RI} : iterações do Algoritmo de Restauração Inexata;

It_{Min} : Iterações da Fase de Minimização;

$n = 5$, $m = 5$ e $q = 3$;

$x_0 = y_0 = z_0 = w_0 = (0, 0, 0, 0, 0)^t$;

$\gamma_0 = (0, 0, 0)$.

f : valor da função objetivo do primeiro nível no ponto obtido pelo algoritmo de Restauração Inexata.

f^* : valor da função objetivo do primeiro nível no ponto obtido em [23].

Em todos os casos tanto os Algoritmos 2.1 quanto 2.1(b) obtiveram sucesso na busca da solução, conforme tabelas 3.5 e 3.6.

3.1.4 Custo de Transporte

O problema de Custo de Transporte [23, 45] pertencem a outra classe que frequentemente é modelada como um problema MPEC. Este problema representa a situação onde o administrador de uma rede tem interesse em maximizar o lucro com a cobrança de

taxas e os usuários visam minimizar os custos de viagem sobre a rede. Segundo [45] este problema pode ser formulado da seguinte maneira:

$$\begin{aligned}
& \underset{t,x}{\text{Maximizar}} \quad \sum_{i \in A_1} t_i x_i \\
& \text{s.a } t_i > -l_i \quad i \in A_1 \\
& \underset{x}{\text{Minimizar}} \quad \sum_{i \in A_1} (c_i + t_i) x_i + \sum_{i \in A_2} c_i x_i \quad \text{s.a} \\
& \sum_{i \in i^+} x_i^{k,l} - \sum_{i \in i^-} x_i^{k,l} = 1 \text{ se } i = k, -1 \text{ se } i = l \text{ e } 0 \text{ c.c} \\
& \forall i \in N, \forall (k, l) \in O \times D \\
& x_i = \sum_{(k,l) \in O \times D} d^{k,l} x_i^{k,l} \quad \forall i \in A \\
& x_i^{k,l} \geq 0 \quad \forall i \in A \quad \forall (k, l) \in O \times D
\end{aligned}$$

Definimos:

c_i : custo fixo para o arco i (não inclui taxas).

N : conjunto dos Nós da rede;

A : conjunto dos índices dos arcos pertencentes à rede;

A_1 : subconjunto de A referente aos arcos que podem estar sujeitos a taxas;

$A_2 = A \setminus A_1$;

i^+ : conjunto dos índices dos arcos partindo do nó i ;

i^- : conjunto dos índices dos arcos chegando no nó i ;

O : conjunto dos índices dos nós de origem;

D : conjunto dos índices dos nós de destino;

x_i^{kl} : fluxo sobre o arco $i \in \{1, 2, \dots, n\}$ da origem k para o destino l ;

d_i^{kl} : proporção de demanda de fluxo entre a origem k e destino l sobre o arco $i \in \{1, 2, \dots, n\}$;

t_i : Taxa cobrada sobre o arco i ;

l_i : Limite para a taxa cobrada sobre o arco i ;

Testamos o algoritmo resolvendo os problemas *Toll1*, *Toll2*, *Toll3*, *Toll4*, *Toll5* propostos em [23].

Critério de Parada do Algoritmo :

$$\|d_{tan}^k\| \leq 10^{-4}, \|C(s^k)\| \leq 10^{-4};$$

Número máximo de iterações externas ($k=250$);

It_{RI} : iterações do Algoritmo de Restauração Inexata;

It_{Min} : Contador da Fase de Minimização;

Constante de Restauração $r = 0.9$;

Critério de Parada do Algoritmo de Projeções : $\|R(y^k)\| \leq 10^{-4}$;

Como pontos iniciais utilizamos $\mu = 0$, $\gamma = 0$ e para as variáveis primais x e y os pontos sugeridos em [23].

Uma vez que em [23] não fica claro quais parâmetros l_i foram utilizados, para efeito de teste supomos $l_i = -1$ para todo i .

Como não é possível comparar as soluções obtidas pelos algoritmos propostos (2.1 e 2.1(b)) com a solução obtida em [23] porque os os parâmetros l_i não estão disponíveis, podemos concluir que ambos algoritmos satisfizeram as condições de parada gerando "pontos estacionários". Mas é importante ressaltar que no teste 1 os pontos estacionários foram diferentes, e o encontrado pelo algoritmo 2.1(b) foi melhor, conforme tabelas 3.7 e 3.8.

Tabela 3.1: Algoritmo 2.1 - Problemas Clássicos

Teste	$\ d_{tan}^k\ $	$\ C(s^k)\ $	It_{RI}	It_{Min}	f	f^*
1	10^{-5}	10^{-6}	4	7	-29.2	-29.2
2	10^{-6}	10^{-5}	8	35	-8.92	-8.92
3	10^{-4}	10^{-4}	6	31	-7.58	-7.58
4	10^{-8}	10^{-6}	7	33	-11.99	-11.99
5	10^{-4}	10^{-5}	8	35	-000.92	-3.6
6	10^{-15}	10^{-5}	2	3	-5.6	-3.92
7	10^{-15}	10^{-4}	3	5	1000	1000
8	10^{-4}	10^{-4}	2	3	81.32	81.32
9	NC	NC				
10	10^{-8}	10^{-8}	4	69	-1	-1
11	10^{-8}	10^{-8}	4	69	-2.2	-2.2
12	10^{-4}	10^{-4}	4	4	-3266	-3266
13	10^{-15}	10^{-8}	4	4	1.5	1.5
14	10^{-8}	10^{-6}	3	5	0.5	0.5
15	10^{-15}	10^{-7}	4	17	17	17
16	10^{-5}	10^{-5}	4	17	9	9
17	10^{-8}	10^{-4}	6	21	0	0
18	10^{-15}	10^{-5}	2	3	5	5

Tabela 3.2: Algoritmo 2.1(b) - Problemas Clássicos

Teste	$\ d_{tan}^k\ $	$\ C(s^k)\ $	It_{RI}	It_{Min}	f	f^*
1	0	10^{-5}	4	7	-29.2	-29.2
2	0	10^{-4}	6	31	-8.92	8.92
3	10^{-4}	10^{-4}	9	57	-7.58	-7.58
4	0	10^{-4}	5	19	-11.99	-11.99
5	10^{-5}	10^{-5}	2	3	-000.92	-3.6
6	0	10^{-6}	2	3	-5.6	-3.92
7	0	10^{-6}	3	15	1000	1000
8	10^{-8}	10^{-4}	4	69	81.32	81.32
9	NC	NC				
10	10^{-6}	10^{-8}	3	68	-1	-1
11	10^{-8}	10^{-8}	4	69	-2.199	-2.199
12	0	10^{-4}	4	4	-3266	-3266
13	0	10^{-8}	4	4	1.5	1.5
14	0	10^{-8}	3	15	0.5	0.5
15	0	10^{-6}	5	29	17	17
16	0	10^{-5}	4	7	9	9
17	10^{-8}	10^{-4}	6	21	0	0
18	0	10^{-5}	2	3	5	5

Tabela 3.3: (Problemas *Bilevel*) Algoritmo 2.1

ρ	n_x	m_y	$Q(\bar{x}, \bar{y})$	$Q(x^*, y^*)$	N.S.G	N.S.L	It_{RI}	It_{Min}
[1.5,3]	4	2	0.89	0.3125	1	3	88	175
[3,3]	4	2	0.8894	0.5	1	3	250	501
[1.2,1.2]	4	2	0.02	0.02	1	3	6	21
[1.3, 2]	4	2	0.4670	0.2725	2	2	27	63
[1.2,1.3]	4	2	0.03250	0.03250	1	3	11	31
[1.2,1.2,1.2,1.2]	6	4	0.04	0.04	1	15	17	53
[1.2,1,1,1]	6	4	5	0.01	1	1	250	550
[1.2,1.2,1.3]	7	3	0.0425	0.0425	1	7	19	59
[1.2,1.2,1.2,1.2,1.2]	10	5	0.05	0.05	1	31	27	53
[1.2,1.2,1.3,1.3,1.2]	10	5	0.075	0.075	1	31	91	201

Tabela 3.4: (Problemas *Bilevel*) Algoritmo 2.1(b)

ρ	n_x	m_y	$Q(\bar{x}, \bar{y})$	$Q(x^*, y^*)$	N.S.G	N.S.L	It_{RI}	It_{Min}
[1.5,3]	4	2	0.89	0.3125	1	3	75	165
[3,3]	4	2	0.8894	0.5	1	3	150	480
[1.2,1.2]	4	2	0.02	0.02	1	3	12	32
[1.3, 2]	4	2	0.4670	0.2725	2	2	35	58
[1.2,1.3]	4	2	0.03250	0.03250	1	3	15	35
[1.2,1.2,1.2,1.2]	6	4	0.04	0.04	1	15	16	54
[1.2,1,1,1]	6	4	5	0.01	1	1	250	540
[1.2,1.2,1.3]	7	3	0.0425	0.0425	1	7	29	48
[1.2,1.2,1.2,1.2,1.2]	10	5	0.05	0.05	1	31	24	38
[1.2,1.2,1.3,1.3,1.2]	10	5	0.075	0.075	1	31	87	177

Tabela 3.5: Problemas de Estruturas de Rede Algoritmo 2.1

Teste	$\ d_{tan}^k\ $	$\ C(s^k)\ $	It_{RI}	It_{Min}	f	f^*
NDP1	10^{-4}	10^{-6}	19	474	300.5	300.5
NDP2	10^{-5}	10^{-5}	23	543	142.89	142.89

Tabela 3.6: Problemas de Estrutura de Rede Algoritmo 2.1(b)

Teste	$\ d_{tan}^k\ $	$\ C(s^k)\ $	It_{RI}	It_{Min}	f	f^*
NDP1	10^{-4}	10^{-4}	15	478	300.5	300.5
NDP2	10^{-4}	10^{-4}	21	545	142.89	142.89

Tabela 3.7: Problema de Custo de Transporte Algoritmo 2.1

Teste	$\ d_{tan}^k\ $	$\ C(s^k)\ $	It_{RI}	It_{Min}	f
Toll1	10^{-5}	10^{-4}	4	3	3.24
Toll2	10^{-6}	10^{-4}	8	203	0
Toll3	10^{-6}	10^{-5}	4	4	0
Toll4	10^{-6}	10^{-5}	4	4	0
Toll5	10^{-6}	10^{-4}	4	4	0

Tabela 3.8: Problema de Custo de Transporte Algoritmo 2.1(b)

Teste	$\ d_{tan}^k\ $	$\ C(s^k)\ $	It_{RI}	It_{Min}	f
Toll1	10^{-5}	10^{-6}	12	207	7
Toll2	10^{-6}	10^{-5}	5	70	0
Toll3	10^{-4}	10^{-4}	4	4	0
Toll4	10^{-5}	10^{-4}	3	3	0
Toll5	10^{-6}	10^{-5}	4	4	0

Capítulo 4

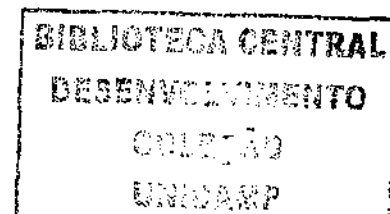
Aplicação ao Problema de Tráfego Urbano

4.1 Introdução

O tráfego congestionado tem se tornado parte do dia-a-dia dos moradores das grandes áreas metropolitanas. Com o crescimento destas áreas nem sempre é possível a construção de novas ruas ou avenidas visando suprir o aumento do fluxo de veículos. Do ponto de vista econômico, segundo pesquisas realizadas nos Estados Unidos [7], são gastos cerca de 48 bilhões de dólares por ano com este tipo de problema e aproximadamente 65% dos motoristas das cidades americanas analisadas já passaram por este problema.

No Brasil, o mesmo problema ocorre nas grandes cidades, em particular na maior cidade do país. Estamos nos referindo a São Paulo. A figura 4.1 descreve um esboço da região onde os problemas com congestionamento são constantes.

Para minimizar os impactos deste problema é necessário que as autoridades respon-



sáveis tomem decisões visando controlá-lo. Uma alternativa seria incorporar impostos ou pedágios (taxas) sobre alguns trajetos (ruas ou avenidas) tentando desviar o trânsito durante alguns períodos do dia.

Na literatura, o problema de determinar pedágios com o objetivo de reduzir o congestionamento é frequentemente chamado de Problema de Custo do Congestionamento, que nesta seção vamos nos referir como **CPP**, motivado pelo termo em inglês "Congestion Pricing Problem".

Em geral o problema CPP pode ser dividido em duas classes. A primeira, conhecida como Primeiro Melhor Problema de Custo de Pedágios ([7], [38] e [39]) assume que todos os arcos que compõem a rede estão sujeitos a pedágios. Neste trabalho vamos nos referir a este problema como **FBTPP**, notação motivada pelo termo em inglês "First-Best Toll Pricing Problem". Já a segunda é conhecida como Segundo Melhor Problema de Custo de Pedágios ([40] e [44]), neste caso considera-se que existem arcos onde não é feita a cobrança de pedágios. Vamos nos referir a este problema como **SBTPP**, notação motivada pelo termo em inglês "Second-Best Toll Pricing Problem".

O problema FBTPP pode ser considerado um caso particular do SBTPP se assumirmos que o conjunto de arcos que não estão sujeitos ao pedágio é vazio.

Muitos pesquisadores durante anos tem desenvolvido modelos e algoritmos para a resolução do SBTPP ([13], [32],[40],[45] e [59]). Em [40] a abordagem proposta explora as propriedades do problema SBTPP e apresenta uma reformulação do problema baseada na literatura do problema MPEC. Neste mesmo trabalho foi considerado um algoritmo de resolução baseado na metodologia de Planos de Cortes [10].

O principal objetivo desta seção é usar a abordagem utilizada em [40] para o caso onde a demanda do problema STBPP é fixa.

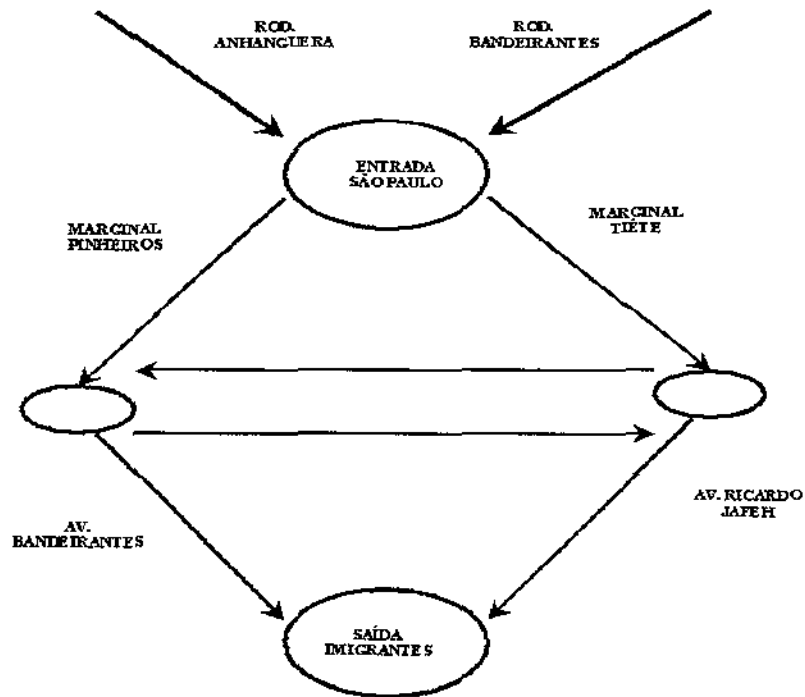


Figura 4.1: Esboço da Região

4.2 Descrição do Problema

4.2.1 O Problema SBTPP com Demanda Elástica

Para facilitar a formulação do problema definimos

i : índice dos arcos da rede;

I : conjunto de todos os índices dos arcos contidos na rede;

k : índice do par Origem/Destino;

K : conjunto de todos os índices dos pares Origem/Destino;

x^k : fluxo do arco para o par Origem/Destino k ;

t_k : demanda para o par Origem/Destino k ;

E_k : $E_k = e_q - e_p$, onde e_i é o vetor canônico;

v : vetor de fluxo agregado, neste caso $v = \sum_{k \in K} x^k$;

$s_i(v)$: custo em tempo de uma unidade de fluxo sobre o arco i ;

$w_k(t)$: função de demanda inversa para o par Origem/Destino k ;

A : matriz de restrição de fluxo;

T_i : pedágio cobrado sobre o arco i ;

Y : conjunto de índices dos arcos que não estão sujeitos a cobrança de pedágios(taxa).

As seguintes hipótese sobre o problema STBPP são assumidas em [40].

Consideremos

$V = \{(v, t) : v = \sum_{k \in K} x^k, Ax^k = E_k t_k, x_k, t_k \geq 0 \quad \forall k \in K\}$ e assumimos que V é limitado;

s_i é contínua e diferenciável $\forall i \in I$;

w_k é contínua $\forall k \in K$;

Do ponto de vista prático estas hipóteses não comprometem a aplicação do problema STBPP [40].

Por outro lado esta hipóteses estão entre as pedidas para o resultado de convergência do Algoritmo de Restauração Inexata.

4.2.2 Formulação MPEC

Podemos definir o problema STBPP como um problema MPEC, conforme [40]:

$$\underset{v, T, t}{\text{Minimizar}} \quad \langle s(v), v \rangle - \sum_{k \in K} \int_0^{t_k} w_k(u) du$$

s.a

$$T_i = 0 \text{ se } i \in Y$$

$$T_i \geq 0 \text{ se } i \notin Y$$

$$\langle s(v) + T, u - v \rangle - \langle w(t), d - t \rangle \geq 0 \quad \forall (u, d) \in V,$$

onde V foi definido acima.

A função objetivo $\langle s(v), v \rangle$ corresponde ao tempo total gasto pelo fluxo sobre a rede ou custo em tempo associado a rede. O conjunto V representa a região viável do segundo nível que requer fluxos viáveis positivos que satisfaçam a restrição de chegada e saída de fluxo nos nós da rede. Já o problema de inequações variacionais parametrizado em T assegura que a rede se mantém em equilíbrio para um T fixo [38].

Como o objetivo deste capítulo é explorar o problema SBTPP com demanda fixa (**SBTPPF**) deixamos para trabalhos futuros o estudo mais detalhado do problema SBTPP com demanda elástica.

O Problema SBTPPF

Seguindo a mesma notação usada para o problema SBTPP com demanda elástica, podemos definir o SBTPP com demanda fixa (SBTPPF) como um problema MPEC da seguinte forma:

Formulação MPEC do SBTPPF

$$\underset{v, T}{\text{Minimizar}} \langle s(v), v \rangle$$

s.a

$$T_i = 0 \text{ se } i \in Y$$

$$T_i \geq 0 \text{ se } i \notin Y$$

$$\langle s(v) + T, u - v \rangle \geq 0 \quad \forall u \in V,$$

onde $V = \{v \in \mathbb{R}^n : v = \sum_{k \in K} x^k, Ax^k = b_k, x_k \geq 0 \forall k \in K\}$ e s_i satisfazem as hipóteses usadas no problema anterior.

Para facilitar o entendimento do problema vamos reescrever o problema acima utilizando a variável v_i que representa o fluxo total agregado sobre o arco i . Assim podemos reescrever o problema acima da seguinte forma:

$$\underset{v, T}{\text{Minimizar}} \langle s(v), v \rangle$$

s.a

$$T_i = 0 \text{ se } i \in Y$$

FD-MPEC

$$T_i \geq 0 \text{ se } i \notin Y$$

$$\langle s(v) + T, v - z \rangle \leq 0 \quad \forall z \in V$$

onde $V = \{z \in \mathbb{R}^n : Az = b \text{ e } z \geq 0\}$, $s_i : \mathbb{R}^n \rightarrow \mathbb{R}$ são pelo menos duas vezes diferenciáveis, separáveis e monótonas, para $i \in \{1, \dots, n\}$.

O exemplo abaixo deixa clara esta reformulação:

Exemplo 4.1. Consideremos a rede abaixo estudada em [41]:

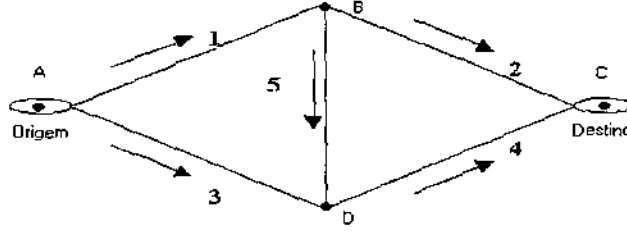


Figura 4.2: Interpretação Geométrica da Rede

Neste caso $K = \{A, B, C, D\}$, $I = \{1, 2, 3, 4, 5\}$ e $Y = I$.

Hipótese de demanda: O fluxo total que sai da origem A deve ser igual a 6 unidades, e além disso o fluxo total que deve chegar ao destino C é exatamente 6 unidades.

Restrições de saída: Na origem A existem dois arcos de saída de fluxo. Levando em consideração a hipótese de demanda, gera-se a seguinte restrição,

$$-v_1 - v_3 = -6, \quad (4.1)$$

ou seja, a quantidade total de fluxo que sai de A deve ser igual 6 unidades.

Restrições dos nós intermediários: Como hipótese supomos que nenhuma unidade de fluxo pode ficar pelo caminho na rede, ou seja, a quantidade de fluxo que chega em um nó diferente da origem ou destino deve ser igual à quantidade que sai deste mesmo nó. Analisando o nó B gera-se a seguinte restrição:

$$v_1 = v_5 + v_2, \quad (4.2)$$

Utilizando um raciocínio análogo para o nó D , obtemos

$$v_3 + v_5 = v_4 \quad (4.3)$$

Restrições de chegada: O destino C está conectado a dois arcos de chegada, levando em consideração a hipótese de demanda, obtemos a seguinte restrição:

$$v_2 + v_4 = 6 \quad (4.4)$$

Utilizando (4.1), (4.2), (4.3), (4.4), podemos definir o conjunto viável do segundo nível do problema MPEC: $V = \{z \in \mathbb{R}^n : Az = b \text{ e } v \geq 0\}$, onde

$$A = \begin{pmatrix} -1 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad (4.5)$$

e

$$b = \begin{pmatrix} -6 \\ 0 \\ 0 \\ 6 \end{pmatrix} \quad (4.6)$$

Propriedades do SBTPPF

Consideremos os seguintes problemas e suas respectivas soluções

$$\begin{aligned} v^s &= \underset{v}{\operatorname{argmin}} \langle s(v), v \rangle \\ \text{s.a } v &\in V \quad S - OPT \end{aligned}$$

Encontrar $v^u \in V$ tal que

$$\langle s(v^u) + T, v^u - z \rangle \leq 0 \quad \forall z \in V, \quad U - OPT$$

onde $V = \{z \in \mathbb{R}^n : Az = b \text{ e } z \geq 0\}$.

Seja (v^*, T^*) uma solução do FD-MPEC, então pode-se estabelecer a seguinte relação entre as soluções dos três problemas (FD-MPEC, S-OPT e U-OPT) [40]:

$$\langle s(v^u), v^u \rangle \geq \langle s(v^*), v^* \rangle \geq \langle s(v^s), v^s \rangle \quad (4.7)$$

Consideremos os seguintes subconjuntos

$$S^* = \{v^* \in \mathbb{R}^n : v^* = \arg \min_{v \in V} \langle s(v), v \rangle\};$$

$$U_T^* = \{\bar{v} \in \mathbb{R}^n : \langle s(\bar{v}) + T, u - \bar{v} \rangle \geq 0 \ \forall u \in V\};$$

$$\Gamma = \{T \in \mathbb{R} : \emptyset \neq U_T^* \subset S^*\};$$

$$W(\bar{v}) = \{\bar{v} \in \mathbb{R}^n : \bar{v} \in U_T^*\}.$$

Teorema 4.1. Se $s_i : \mathbb{R}^n \rightarrow \mathbb{R}$ são pelo menos duas vezes diferenciáveis, estritamente monótonas crescente, para $i \in \{1, \dots, n\}$, $\langle s(v), v \rangle$ é estritamente convexa e $\bar{v}$ é a única solução do problema S-OPT. Então

$$\Gamma = W(\bar{v}) \quad (4.8)$$

Prova: ver Teorema 3 [38] . ■

Teorema 4.2. Dado o seguinte problema

$$\begin{aligned} & \underset{v, T}{\text{Minimizar}} \quad \langle s(v), v \rangle \\ & \text{s.a} \quad \langle s(v) + T, v - z \rangle \leq 0 \ \forall z \in V \end{aligned}$$

onde $V = \{z \in \mathbb{R}^n : Az = b \text{ e } z \geq 0\}$, $s_i : \mathbb{R}^n \rightarrow \mathbb{R}$ são pelo menos duas vezes diferenciáveis, estritamente monótonas, para $i \in \{1, \dots, n\}$ e $\langle s(v), v \rangle$ é estritamente convexa. Então o conjunto de soluções do problema acima é exatamente $\Upsilon = \{(v^*, T) : T \in \Gamma\}$.

Prova: ver Teorema 7 [38]. ■

Os Teoremas 4.1 e 4.2 valem quando consideramos $T \geq 0$. A relação (4.7) também é válida para o problema SBTPP com demanda elástica.

Por fim, é interessante ressaltar que em geral as unidades de T estão em minutos, mas este fato na prática não influencia muito, pois supondo que o administrador da rede cobre de um arco i a quantia de M_i ($R\$$) unidades monetárias e se R ($\frac{R\$}{min}$) é o valor de uma unidade de fluxo por minuto, então $T_i = \frac{M_i}{R}$. Além disso se T_i for negativo significa que o administrador da rede pode estar sugerindo que seja dado um subsídio aos usuários da rede para que seja utilizado o arco i .

Na próxima seção vamos analisar qual é a influência da variável T na rede.

4.2.3 Sensibilidade ao Pedágio

O objetivo desta seção é avaliar a influência do pedágio sobre os arcos da rede. Vamos analisar diferentes casos, mas utilizaremos o mesmo exemplo de fluxo da figura 4.2.2 e as seguintes funções s_i :

Funções de Fluxo [41]: $s_1(v_1) = 10v_1$, $s_2(v_2) = v_2 + 50$, $s_3(v_3) = v_3 + 50$, $s_4(v_4) = 10v_4$ e $s_5(v_5) = v_5 + 10$.

Consideremos o problema VIP parametrizado em T :

$$\langle s(v) + T, v - z \rangle \leq 0 \quad \forall z \in V,$$

Para resolver o problema acima utilizamos o algoritmo baseado em projeções proposto por Solodov e Svaiter [66].

Caso I: Se $T = 0$, resolvendo o problema (4.2.3) acima obtemos a solução $v_I^* =$

$$(4, 2, 2, 4, 2)^t \text{ e } \langle s(v_I^*), v_I^* \rangle = 552;$$

Caso II: Se considerarmos $T_1 = T_2 = \dots = T_4 = 0$ e $T_5 = 10$ temos como solução

$$v_{II}^* = (3.23, 2.76, 2.76, 3.23, 0.47)^t \text{ e } \langle s(v_{II}^*), v_{II}^* \rangle = 504;$$

Caso III: Se considerarmos $T_1 = T_2 = \dots = T_4 = 0$ e $T_5 = 13$ temos como solução

$$v_{III}^* = (3, 3, 3, 3, 0)^t \text{ e } \langle s(v_{III}^*), v_{III}^* \rangle = 498 \text{ (Solução Ótima do Problema MPEC)};$$

Caso IV: Se considerarmos $T_1 = T_2 = \dots = T_4 = 0$ e $T_5 = 10^6$ temos como solução

$$v_{IV}^* = (3, 3, 3, 3, 0)^t;$$

Observe que conforme o valor de T_5 aumenta, o fluxo no arco 5 diminui migrando para os outros arcos. Outro fato interessante é que a mesma solução ótima v pode estar associada, a vetores T diferentes (casos III e IV).

Continuando com a nossa análise, estamos agora interessados em desviar o fluxo na rede em função dos pedágios (T).

Caso V: Se $T = (0, 0, 10^6, 0, 0, 0)^t$ temos a solução $v_V^* = (6, 2.1666, 0, 3.84, 3.84)^t$ e

$$\langle s(v_V^*), v_V^* \rangle = 673.63;$$

Caso VI: Se $T = (0, 10^6, 0, 0, 0, 0)^t$ temos a solução $v_{VI}^* = (2.333, 0, 3.666, 6, 2.333)^t$ e

$$\langle s(v_{VI}^*), v_{VI}^* \rangle = 945.55.$$

Observe que tanto no caso V como no VI quando o valor do pedágio é muito alto, o fluxo é desviado para outros arcos.

4.2.4 Modelagem do Problema SBTPPF

Consideremos o problema de Equilíbrio com Fluxo Limitado [37]

$$\begin{aligned}
 & \underset{v}{\text{Minimizar}} \quad \sum_{i=1}^n \int_0^{v_i} s_i(t) dt & (4.9) \\
 \text{s.a} \quad & v_i \leq c_i \quad (\text{capacidade}) \quad i = 1, \dots, n \quad (BFEP) \\
 & v \in V = \{z \in \mathbb{R}^n : Az = b \text{ e } z \geq 0\}
 \end{aligned}$$

Se c_i fosse conhecido para cada arco i o problema SBTPPF se reduziria a resolver o problema BFEP, mas em geral não é fácil estimar c_i . Esta justificativa não está explícita em [40] onde o problema SBTPP é tratado como um problema MPEC.

Decidimos dar uma interpretação própria da modelagem do problema SBTPPF como um MPEC.

Consideremos as condições de otimalidade do problema BFEP (C.O.T).

$$\begin{aligned}
 \langle s(v) + T, v - u \rangle & \leq 0 \quad \forall u \in V & (4.10) \\
 T_i(c_i - v_i) & = 0 \quad i = 1, \dots, n \quad (C.O.T) \\
 v_i, T_i & \geq 0 \quad i = 1, \dots, n
 \end{aligned}$$

Observe que:

- se $T_i > 0$ então $c_i = v_i$;
- se $T_i = 0$ então $c_i = \infty$;

De maneira intuitiva podemos concluir que a variável T_i faz o papel do pedágio no arco i , com o objetivo de aumentar ou diminuir o fluxo neste arco.

Como c_i existe, mas é desconhecida e na prática estamos interessados em encontrar

o par (T, v) que resolva o problema $C.O.T$ e ao mesmo tempo minimize o custo total de tempo $\langle s(v), v \rangle$, a formulação matemática é:

$$\begin{aligned}
& \underset{v, T}{\text{Minimizar}} \langle s(v), v \rangle \\
& \text{s.a} \\
& T_i = 0 \text{ se } i \notin Y \\
& T_i \geq 0 \text{ se } i \in Y \\
& \langle s(v) + T, v - u \rangle \leq 0 \quad \forall z \in V
\end{aligned}$$

4.3 Resolução via Algoritmo de Restauração Inexata

Para resolver os problemas testes relacionados com o problema SBTPPF utilizamos os Algoritmo 2.1 e 2.1(b).

Mas com o objetivo de acelerar a convergência do algoritmo e além disso encontrar uma solução que do ponto de vista econômico privilegie os usuários da rede, inicializamos a fase de minimização do Algoritmo 2.1 com um ponto inicial $s_0 = (T_0, v_0, \lambda_0, z_0)$ definido da seguinte forma:

$$v_0 = \tilde{v}_k \text{ (ponto restaurado)};$$

$$(T_0, \lambda_0, z_0) \text{ solução do problema de programação não linear convexo:}$$

$$\underset{T, \mu, \gamma}{Minimizar} \quad \frac{1}{2} \|T\|^2$$

$$\text{s.a}$$

$$T_i = 0 \text{ se } i \in Y$$

$$T_i \geq 0 \text{ se } i \notin Y$$

$$s(v_0) + T + A^t \mu - \gamma = 0$$

$$v_0^t \gamma = 0$$

$$\gamma \geq 0$$

Para visualizar os impactos desta alteração consideremos o exemplo da figura 4.2.2. Quando utilizamos s_0 conforme definido acima encontramos a solução $v^* = (3, 3, 3, 3, 0)^t$ e $T^* = (0, 0, 0, 0, 13)^t$, se s_0 é exatamente o ponto obtido na fase de restauração (ponto restaurado $\bar{s}_k$) obtemos como solução $\bar{v} = v^*$ mas $\bar{T} = (4.46, 0, 0, 4.46, 8.58)^t$. Fica claro que $\langle v^*, T^* \rangle \leq \langle v^*, \bar{T} \rangle$.

É importante ressaltar que a incorporação deste procedimento não muda os resultados de convergência obtidos neste trabalho.

4.4 Resultados Numéricos

Critérios de parada e parâmetros de entrada e notações:

$$\|d_{tan}^k\| \leq 10^{-4}, \|C(s^k)\| \leq 10^{-4};$$

Número máximo de iterações externas (k=250);

Constante de Restauração $r = 0.9$;

Critério de Parada do Algoritmo de Projeções : $\|R(y^k)\| \leq 10^{-4}$;

It_{RI} : iterações do Algoritmo de Restauração Inexata;

It_{Min} : Contador de iteração da Fase de Minimização;

T_0, v_0, z_0, w_0 e γ_0 : vetores nulos com suas respectivas dimensões.

Obs. z_0 e w_0 representam μ_0 .

Descrição dos Problemas Testes

Para testar e validar o algoritmo proposto neste trabalho resolvemos alguns testes da literatura e além disso criamos novos testes baseados nos propostos na literatura.

Teste I

O exemplo 4.2.2 e as funções fluxo utilizadas na análise de sensibilidade da variável T .

Teste II

Neste teste propomos uma nova rede a ser estudada, apesar de não estar na literatura tanto a rede quanto as função de fluxo podem ser utilizadas na vida real.

Neste caso definimos a função de custo da seguinte forma :

$$s_i(v_i) = 0.5v_i + i \text{ para todo } i \in A, \quad (4.11)$$

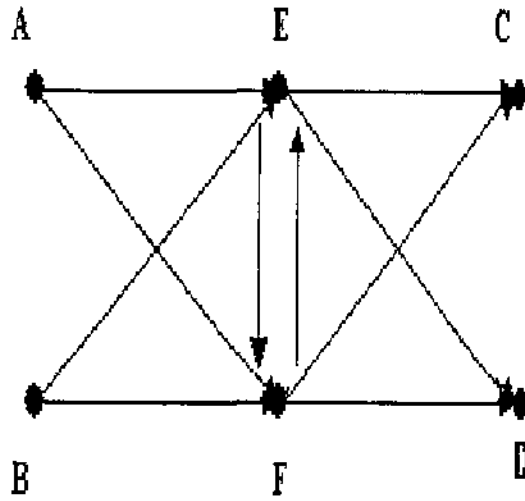


Figura 4.3: Ilustração para a Rede do Teste II

e a mesma tabela de demanda do Teste III.

Testes III-VI

O teste III foi considerado como FBTPPF em [38] e SBTPPF em [40].

Já os parâmetros dos demais testes foram extraídos de [37], mas nunca foram tratados como FBTPPF ou SBTPPF.

A tabela abaixo descreve os parâmetros de entrada dos problemas associados à rede 4.4 e as demandas em cada origem e destino ver ([37]).

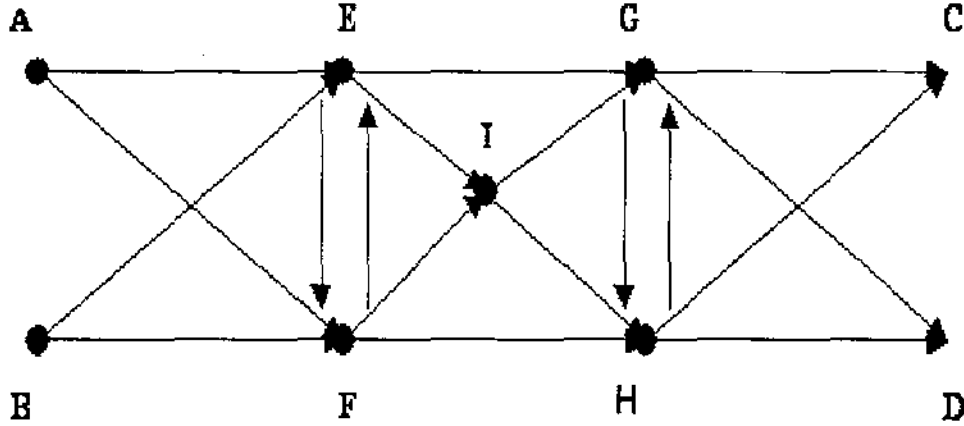


Figura 4.4: Ilustração para a Rede dos Testes III-VI

Neste caso definimos a função de custo da seguinte forma :

$$s_i(v_i) = t_i(1 + 0.15(\frac{v_i}{c_i})^4) \text{ para todo } i \in A \quad (4.12)$$

As tabelas contendo os experimentos numéricos foram organizadas em duas classes, os problemas First Best Toll Pricing Problem e Second Best Toll Pricing Problem. Além disso resolvemos os experimentos utilizando o Algoritmo 2.1(b) proposto na seção de experimentos numéricos.

First Best Toll Pricing Problem

Comparando as soluções obtidas em [41] e [37] nos testes I e III respectivamente, podemos concluir que os valores da solução na função objetivo são iguais aos obtidos pelo Algoritmo 2.1 conforme a Tabela 4.2.

Nos demais testes, como são novos problemas gerados podemos só destacar que foram satisfeitos os critérios de parada do algoritmo em um número razoável de iter-

ações.

O mesmo acontece com o Algoritmo 2.1(b), conforme Tabela 4.3.

Analisando a Tabela 4.4 podemos comparar somente a solução obtida no teste III, neste caso o valor da função objetivo obtido em [40] é 2455.06, já o nosso algoritmo obteve 2521.8.

Em relação às demais, mais uma vez encontramos uma solução que satisfaz os critérios de parada do algoritmo.

De novo o Algoritmo 2.1(b) teve um desempenho semelhante ao algoritmo 2.1, conforme tabela 4.5

* O Arco 1 não cobra pedágio;

** Nenhum arco cobra pedágio;

*** Somente os arcos 3 e 18 cobram pedágio.

Análise da solução obtida

Para fins de análise comparativa, vamos analisar a solução obtida pelo Algoritmo 2.1 com a solução obtida em [37] quando consideramos o problema FBTPPF, conforme Tabela 4.6.

A solução v obtida neste caso é exatamente a solução obtida em [37]. Já a solução T é ligeiramente diferente mas $T^T v$ é exatamente igual. É esperado que aconteça isso pois para uma única solução v podemos ter várias soluções T .

Parâmetros de Entrada da Rede [37]

A tabela 4.1 contém os parâmetros de entrada para os testes III, IV, V e VI.

Tabela 4.1: Parâmetros de Entrada

		Teste III		Teste IV		Teste V		Teste VI	
Link	n.Link	t_i	c_i	t_i	c_i	t_i	c_i	t_i	c_i
A-E	1	5	12	5	10^2	5	25	5	25
A-F	7	6	18	5	16^2	5	25	5	25
B-E	8	3	35	3	35	5	25	5	25
B-F	4	9	35	9	18^2	5	25	5	25
E-F	9	9	20	9	20	5	25	5	25
E-G	2	2	11	2	11	5	25	5	25
E-I	12	8	26	8	26	5	25	5	25
F-E	10	11	4	11	20	5	25	5	25
F-H	5	6	33	11	20	5	25	5	25
F-I	11	7	32	7	32	5	25	5	25
G-C	3	3	25	3	25	5	25	5	25
G-D	18	6	24	6	24	5	25	5	25
G-H	15	9	19	9	19	5	25	5	25
H-C	17	8	39	8	39	5	25	5	25
H-D	6	6	43	6	43	5	25	5	25
H-G	16	4	36	4	36	5	25	5	25
I-G	13	4	26	4	26	5	25	5	25
I-H	14	8	30	8	30	5	25	5	25

Seguem abaixo as tabelas de demandas para os Testes III, IV, V, VI.

Tabelas de Demandas

III	C	D	IV	C	D
A	10	20	A	10	20
B	30	40	B	30	40

V	C	D	VI	C	D
A	25	25	A	10	20
B	25	25	B	30	40

Resultados Numéricos

Tabela 4.2: Resultados Numéricos FBTPPF-Algoritmo 2.1

Teste	$\ d_{tan}^k\ $	$\ C(s^k)\ $	It_{RI} (k)	It_{Min}	f	$\ T\ $
I	10^{-5}	10^{-5}	5	5	498	13
II	10^{-5}	10^{-5}	24	24	4108.83	3.53
III	10^{-5}	10^{-5}	17	82	2253.9	14.3
IV	10^{-5}	10^{-4}	30	650	2160.9	14.233
V	10^{-5}	10^{-5}	5	70	2128.0	5.6569
VI	10^{-5}	10^{-5}	6	71	1967.8	5.6569

Tabela 4.3: Resultados Numéricos FBTPPF-Algoritmo 2.1(b)

Teste	$\ d_{tan}^k\ $	$\ C(s^k)\ $	It_{RI} (k)	It_{Min}	f	$\ T\ $
I	10^{-5}	10^{-4}	6	5	498	13
II	10^{-5}	10^{-4}	22	22	4108.83	3.53
III	10^{-5}	10^{-4}	9	9	2253.9	14.3
IV	10^{-5}	10^{-4}	15	275	2160.9	14.233
V	10^{-5}	10^{-4}	6	71	2128.0	5.6569
VI	10^{-5}	10^{-4}	6	71	1967.8	5.6569

Tabela 4.4: Resultados Numéricos SBTPPF-Algoritmo 2.1

Teste	$\ d_{tan}^k\ $	$\ C(s^k)\ $	It_{RI} (k)	It_{Min}	f	$\ T\ $
I*	10^{-5}	10^{-5}	5	5	498	13.8512
I**	10^{-5}	10^{-5}	3	3	552	0
III***	10^{-5}	10^{-4}	6	6	2521.8	0.2453
IV***	10^{-5}	10^{-4}	8	138	2457.4	1.052
V***	10^{-5}	10^{-4}	3	3	2313.7	0
VI***	10^{-5}	10^{-4}	14	409	2153.3	0

Tabela 4.5: Resultados Numéricos SBTPPF-Algoritmo 2.1(b)

Teste	$\ d_{tan}^k\ $	$\ C(s^k)\ $	It_{RI} (k)	It_{Min}	f	$\ T\ $
I*	10^{-5}	10^{-4}	4	5	498	13.8512
I**	10^{-4}	10^{-4}	3	3	552	0
III***	10^{-5}	10^{-4}	5	5	2521.8	0.2453
IV***	10^{-5}	10^{-4}	8	138	2457.4	1.052
V***	10^{-5}	10^{-4}	3	3	2313.7	0
VI***	10^{-5}	10^{-4}	9	204	2153.3	0

Tabela 4.6: Solução Teste III FBTPPF

		Solução RI		paper [37]	
Link	n.Link	v_i	T_i	v_i	T_i
A-E	1	9.41	0	9.41	0
A-F	7	20.6	0	20.6	0
B-E	8	38.34	4	38.34	4
B-F	4	31.67	0	31.67	0
E-F	9	0	0	0	0
E-G	2	21.3	10.34	21.3	11.2
E-I	12	26.44	0	26.44	0
F-E	10	0	0	0	0
F-H	5	39.47	7.2	39.47	7.2
F-I	11	12.78	0	12.78	0
G-C	3	29.6	4.86	29.6	4
G-D	18	20.76	0.86	20.76	0
G-H	15	0	0	0	0
H-C	17	19.39	0	19.39	0
H-D	6	39.24	0	39.24	0
H-G	16	0	0	0	0
I-G	13	29.06	2.34	29.06	3.2
I-H	14	10.16	0	10.16	0
$s(v)^tv$		2253.917		2253.917	
T^tv		887.57		887.57	

Tabela 4.7: Solução Teste III SBTPPF

Link	n.Link	Solução RI	
		v_i	T_i
A-E	1	5.055	0
A-F	7	29.944	0
B-E	8	49.85	0
B-F	4	20.15	0
E-F	9	0	0
E-G	2	21.3	0
E-I	12	27.16	0
F-E	10	0	0
F-H	5	45.09	0
F-I	11	27.16	0
G-C	3	40	0.00018
G-D	18	14.9	0.244
G-H	15	0	0
H-C	17	0	0
H-D	6	45.09	0
H-G	16	0	0
I-G	13	29.06	0
I-H	14	10.16	0
$s(v)^t v$	2521.74		
$T^t v$	3.66		

Capítulo 5

Conclusões e trabalho futuro

Mostramos nesta tese a viabilidade de resolver uma classe de problemas MPEC utilizando a estratégia de restauração inexata.

Os resultados teóricos se aplicam a uma classe bastante ampla, compreendendo os problemas onde as funções que definem a região viável devem ser duas vezes continuamente diferenciáveis e as restrições parametrizadas dos problemas do segundo nível devem definir regiões convexas. Também mostramos que a estratégia apresentada se aplica a problemas MPCC, onde a convexidade da região definida pelas funções não é necessária.

Este trabalho é uma continuação natural da tese [18], onde o mesmo enfoque foi usado na resolução de problemas de otimização em dois níveis. Alguns resultados obtidos nessa tese foram utilizados neste trabalho, e novas provas foram feitas para parte desses resultados devido à melhor compreensão de várias questões ao aprofundar o estudo do problema.

O objetivo fundamental para usar este enfoque é o de preservar a estrutura em

dois níveis. O único trabalho na literatura que conhecemos, que preserva esta estrutura em algum sentido é o recente artigo de [24]. É interessante mencionar que não foi obtido nenhum resultado teórico de convergência para o algoritmo proposto em [24], confirmando a dificuldade da análise teórica deste tipo de problemas.

O primeiro resultado importante obtido nesta tese é que com hipóteses fracas sobre o problema é garantida a viabilidade dos pontos de acumulação da sequência gerada pelo algoritmo. Este resultado permite relaxar as exigências no passo de restauração no algoritmo apresentado em [50].

O teorema de convergência é enunciado de forma diferente do feito nos trabalhos anteriores. O enunciado novo pode dar a sensação de que é imposta uma condição ao algoritmo, mas para suavizar esta sensação, mostramos que existem hipóteses (fortes) sobre o problema, que garantem a verificação das propriedades exigidas ao algoritmo.

Em relação à otimalidade, o teorema provado dá condições suficientes para que exista um ponto de acumulação viável e que verifique a condição AGP.

Este resultado é relevante no caso em que as soluções dos problemas do segundo nível verificam a complementaridade estrita. Neste caso, como no caso em que os problemas do segundo nível não possuem restrições de desigualdade, o problema MPEC resulta diferenciável e neste caso existem pontos de acumulação viáveis e KKT do MPEC. Este resultado também justifica teoricamente os algoritmos usados para resolver o MPEC, onde os problemas do segundo nível são resolvidos "exatamente" com precisão ϵ .

No artigo [5] é mostrado que sob certas condições, das quais a mais importante é a complementaridade estrita, as soluções do MPEC são pontos KKT de uma variação do problema reformulado com o sistema KKT associado, que resulta da retirada das restrições de complementaridade e substituídas por $g(x, y) + \gamma \geq 0$. Este resultado se aplica ao nosso problema. Quando a trajetória das soluções do problema do segundo

nível é o gráfico de uma função diferenciável do parâmetro (o caso mais favorável para o MPEC estudado), o MPEC pode ser pensado como um problema de um único nível, diferenciável. Neste caso a regularidade do problema não está afetada pelas restrições de complementaridade, e nos parece que há alguma relação entre este fato e o resultado mencionado. Este é um assunto que nos interessa pesquisar, especialmente a relação entre esta trajetória e os sistemas KKT associados ao segundo nível.

Nos artigos [73, 43] são introduzidas relaxações ao sistema KKT associado ao MPEC ou ao MPCC, e se define uma qualificação para as restrições restantes denotada MPEC-LICQ, que consiste na independência linear destas. Com base nesta qualificação se definem condições de otimalidade para o MPEC e o MPCC. Queremos pesquisar futuramente se é possível adaptar o enfoque de restauração inexata usando estas idéias.

Do ponto de vista da prática e das aplicações, a estratégia usada para resolver os problemas é original. Implementamos o algoritmo usando um algoritmo de projeções para a fase de restauração. Este algoritmo requer a convexidade da região de viabilidade do segundo nível e é eficiente quando esta região é definida por funções afins. Os experimentos numéricos apresentados mostram que o método implementado é bom. De qualquer forma, enfatizamos que há liberdade total para escolher o algoritmo adequado para a fase de restauração e é importante considerar o que é conveniente em cada caso particular.

Nas aplicações ao problema de tráfego também obtivemos resultados alentadores. Este problema é especialmente apto para ser tratado com o algoritmo de projeções. Combinar este algoritmo com programação inteira será assunto de trabalho futuro.

Os resultados de sensibilidade e estabilidade de problemas parametrizados são essenciais e devemos aprofundar o estudo destes tópicos, que são cheios de detalhes e questões bastante difíceis. Ver [11].

Para finalizar, ressaltamos que este enfoque nos permitiu obter resultados de convergência tanto para o problema *bilevel* em [18] como para o MPEC e o MPCC nesta tese.

Capítulo 6

Apêndice

6.1 Limitações para o Erro

Teorema 6.1. (Teorema da Função Implícita) Seja $F : \mathbb{R}^k \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ contínuo e diferenciável (C^1). Suponhamos que $F(x_0, y_0) = 0$ e

$$\det[F'_y(x_0, y_0)] \neq 0. \quad (6.1)$$

Então existe um aberto $W \subset \mathbb{R}^k$ e $w : W \rightarrow \mathbb{R}^m$ contínua e diferenciável tais que

1. $x_0 \in W$ e $w(x_0) = y_0$;
2. $F(x, w(x)) = 0 \ \forall \ x \in W$.

Prova: Ver [61].

As demonstrações dos lemas a seguir foram baseadas no Lema 3.2.4 e Teorema 3.2.2 de [33].

Lema 6.1. $C'_{y,\mu,\gamma}(x, y(x), \mu(x), \gamma(x))$ é não singular para todo $x \in X$, onde $(x, y(x), \mu(x), \gamma(x))$ é a solução do sistema (2.4).

Prova:

Pela definição de $C(x, y, \mu, \gamma)$ conforme (2.2) temos:

$$C'_{y,\mu,\gamma}(x, y, \mu, \gamma) = \begin{pmatrix} H & A^T & -B^T \\ A & 0 & 0 \\ \text{diag}(\gamma) * B & 0 & \text{diag}(g(x, y(x))) \end{pmatrix}$$

onde $H = [\nabla_y G(x, y) - \sum_{i=1}^l \nabla_y^2 g_i(x, y) \gamma_i]$, $A = h'_y(x, y)$ é a matriz jacobiana das restrições afins, $B = g'_y(x, y)$ é a matriz jacobiana das restrições de desigualdade e $\text{diag}(z) \in \mathbb{R}^{n_y \times n_y}$ é uma matriz diagonal, com componentes z_i .

Sejam $u \in \mathbb{R}^m$, $v \in \mathbb{R}^q$ e $w \in \mathbb{R}^l$ tais que:

$$C'_{y,\mu,\gamma}(x, y(x), \mu(x), \gamma(x)) \begin{pmatrix} u \\ v \\ w \end{pmatrix} = 0$$

ou seja,

$$Hu + A^T v - B^T w = 0 \quad (6.2)$$

$$Au = 0 \quad (6.3)$$

$$\gamma_i \nabla_y^T g_i(x, y(x))u + g_i(x, y(x))w_i = 0 \text{ para } i = 1, \dots, l. \quad (6.4)$$

Suponhamos, sem perda de generalidade que $g_i(x, y(x)) = 0$ para $i = 1, \dots, p$ e $g_i(x, y(x)) > 0$ para $i = p + 1, \dots, l$ e, portanto pela hipótese A4:

$$\gamma_i(x) = 0 \text{ para } i = p + 1, \dots, l. \quad (6.5)$$

$$\gamma_i(x) > 0 \text{ para } i = 1, \dots, p. \quad (6.6)$$

Usando (6.5), (6.6) e (6.4) temos :

$$\nabla_y^T g_i(x, y(x))u = 0 \text{ para } i = 1, \dots, p. \quad (6.7)$$

$$w_i = 0 \text{ para } i = p + 1, \dots, l. \quad (6.8)$$

Multiplicado a equação (6.2) por u^T temos:

$$u^T H u + u^T A^T v - u^T B^T w = 0$$

Utilizando (6.3), (6.7) e (6.8) vem

$$u^T H u = 0.$$

Como (6.3) e (6.7) implicam que $u \in Nu(h'_y(x, y(x)) \cap Nu((g_I)'_y(x, y(x)))$, onde I é o conjunto de índices das restrições ativas em $(x, y(x))$ pela hipótese (A3) temos,

$$u = 0.$$

Assim, reescrevendo o sistema temos:

$$A^T v - B^T w = 0 \quad (6.9)$$

$$\text{diag}(g(x, y(x)))w = 0 \quad (6.10)$$

Podemos escrever (6.9) como:

$$(A^T \quad -B^T) \begin{pmatrix} v \\ w \end{pmatrix} = 0$$

ou ainda por (6.8)

$$(A^T \quad -B_p^T) \begin{pmatrix} v \\ w_p \end{pmatrix} = 0$$

onde $B_p \in R^{m \times p}$ é uma parte da matriz B associada as restrições ativas e w_p é o vetor de dimensão p que corresponde às variáveis não nulas de w .

Pela hipótese (A1) a matriz $(A^T \quad -B_p^T)$ tem posto completo e portanto

$$v = 0 \quad w_p = 0.$$

Concluimos que $(u, v, w) = (0, 0, 0)$ e portanto $C'_{y,\mu,\gamma}(x, y(x), \mu(x), \gamma(x))$ é não singular para todo $x \in X$.

■

Lema 6.2. (*Limitação local para o erro*):

Seja $x \in X$ dado e $(y(x), \mu(x), \gamma(x))$ tal que $C(x, y(x), \mu(x), \gamma(x)) = 0$. Se $\mu(x)$ e $\gamma(x)$ são funções contínuas, então existem $\beta > 0$ e $\varepsilon > 0$ tais que

$$\|(y, \mu, \gamma) - (y(x), \mu(x), \gamma(x))\| \leq \beta \|C(x, y, \mu, \gamma)\|$$

para todo $(y, \mu, \gamma) \in V_{y,\mu,\gamma}((y(x), \mu(x), \gamma(x)), \varepsilon)$.

Prova:

Como $C(x, y, \mu, \gamma)$ é continuamente diferenciável e

$$C'_{y,\mu,\gamma}(x, y(x), \mu(x), \gamma(x))$$

é não singular, pelo teorema da função inversa [61] existe $\varepsilon_1 > 0$ tal que

$$C(x, y, \mu, \gamma)$$

possui uma inversa diferenciável G em $V_{y,\mu,\gamma}((x, y(x), \mu(x), \gamma(x)), \varepsilon_1)$. Então

$$G(C(x, y, \mu, \gamma)) = (x, y, \mu, \gamma) \quad (6.11)$$

e

$$G'(C(x, y, \mu, \gamma)) = C'_{y,\mu,\gamma}((x, y, \mu, \gamma))^{-1} \quad (6.12)$$

para $(x, y, \mu, \gamma) \in V_{y,\mu,\gamma}((x, y(x), \mu(x), \gamma(x)), \varepsilon_1)$.

Como X é compacto, $y(x)$, $\mu(x)$ e $\gamma(x)$ são uniformemente contínuas então ε_1 não depende de x .

Expandindo G em série de Taylor, em torno de $C(x, y(x), \mu(x), \gamma(x))$, obtemos:

$$\begin{aligned} G(C(x, y, \mu, \gamma)) &= G(C(x, y(x), \mu(x), \gamma(x))) + \\ G'(C(x, y(x), \mu(x), \gamma(x))) &(C(x, y, \mu, \gamma) - C(x, y(x), \mu(x), \gamma(x))) + \\ R(C(x, y, \mu, \gamma) - C(x, y(x), \mu(x), \gamma(x))) \end{aligned}$$

onde

$$\lim_{C(x,y,\mu,\gamma) \rightarrow C(x,y(x),\mu(x),\gamma(x))} \frac{R(C(x, y, \mu, \gamma) - C(x, y(x), \mu(x), \gamma(x)))}{\|C(x, y, \mu, \gamma) - C(x, y(x), \mu(x), \gamma(x))\|} = 0$$

Utilizando (6.11) e (6.12) e o fato de que $C(x, y(x), \mu(x), \gamma(x)) = 0$, temos que

$$(x, y, \mu, \gamma) = (x, y(x), \mu(x), \gamma(x)) + C'(x, y(x), \mu(x), \gamma(x))^{-1} C(x, y, \mu, \gamma) + R(C(x, y, \mu, \gamma))$$

E portanto

$$(x, y, \mu, \gamma) - (x, y(x), \mu(x), \gamma(x)) = C'(x, y(x), \mu(x), \gamma(x))^{-1} C(x, y, \mu, \gamma) + R(C(x, y, \mu, \gamma))$$

onde

$$\lim_{C(x, y, \mu, \gamma) \rightarrow 0} \frac{R(C(x, y, \mu, \gamma))}{\|C(x, y, \mu, \gamma)\|} = 0$$

Calculando a norma temos:

$$\|(y, \mu, \gamma) - (y(x), \mu(x), \gamma(x))\| \leq \|C'(x, y(x), \mu(x), \gamma(x))^{-1}\| \|C(x, y, \mu, \gamma)\| + \|R(C(x, y, \mu, \gamma))\| \quad (6.13)$$

Agora, dado $\alpha > 0$, pela definição de R , existe $\varepsilon_2 > 0$ tal que

$$\|C(x, y, \mu, \gamma)\| \leq \varepsilon_2 \Rightarrow \left\| \frac{R(C(x, y, \mu, \gamma))}{\|C(x, y, \mu, \gamma)\|} \right\| \leq \alpha.$$

Como C é contínua e $C(x, y(x), \mu(x), \gamma(x)) = 0$, existe $\varepsilon_3 > 0$ tal que

$$\|C(x, y, \mu, \gamma)\| \leq \varepsilon_2 \quad (6.14)$$

para todo $(y, \mu, \gamma) \in V_{y, \mu, \gamma}((y(x), \mu(x), \gamma(x)), \varepsilon_3)$.

Assim, dado $\alpha > 0$

$$\|R(C(x, y, \mu, \gamma))\| \leq \alpha \|C(x, y, \mu, \gamma)\|$$

para todo

$$(x, y, \mu, \gamma) \in V_{y, \mu, \gamma}(y(x), \mu(x), \gamma(x), \varepsilon_3) \cap V_{y, \mu, \gamma}((y(x), \mu(x), \gamma(x)), \varepsilon_1)$$

Utilizando (6.14), a equação (6.13) pode ser escrita como:

$$\begin{aligned} & \|(y, \mu, \gamma) - (y(x), \mu(x), \gamma(x))\| \leq \\ & |C'(y(x)(x, , \mu(x), \gamma(x))^{-1}| \|C(x, y, \mu, \gamma)\| + \alpha \|C(x, y, \mu, \gamma)\| \end{aligned}$$

para todo

$$(x, y, \mu, \gamma) \in V_{y, \mu, \gamma}((y(x), \mu(x), \gamma(x)), \varepsilon_3) \cap V_{y, \mu, \gamma}((y(x), \mu(x), \gamma(x)), \varepsilon_1)$$

ou seja,

$$\begin{aligned} & \|(y, \mu, \gamma) - (y(x), \mu(x), \gamma(x))\| \leq \\ & (|C'(x, y(x), \mu(x), \gamma(x))^{-1}| + \alpha) \|C(x, y, \mu, \gamma)\| \end{aligned}$$

para todo

$$(x, y, \mu, \gamma) \in V_{y, \mu, \gamma}((y(x), \mu(x), \gamma(x)), \varepsilon_3) \cap V_{y, \mu, \gamma}((y(x), \mu(x), \gamma(x)), \varepsilon_1)$$

A continuidade uniforme das funções envolvidas garante que podemos escolher ε_2 e ε_3 independentemente de x . Assim, tomando $\varepsilon = \min\{\varepsilon_3, \varepsilon_1\}$ e

$$\beta \geq (|C''_{y,\mu,\gamma}((x, y(x), \mu(x), \gamma(x))^{-1}| + \alpha) > 0 \quad (6.15)$$

temos o resultado desejado.

■

Bibliografia

- [1] R. Andreani, A. Friedlander e J. M. Martínez, *On the Solution of Finite-Dimensional Variational Inequalities Using Smooth Optimizations with Simple Bounds*, Journal on Optimization Theory and Applications 94, pp. 635-657, 1997.
- [2] R. Andreani, J. M. Martínez e B. F. Svaiter, *On the Regularization of mixed complementarity problems*, Numerical Functional Analysis and Optimization, 21 (5 & 6), pp. 589-600, 2000.
- [3] R. Andreani e J. M. Martínez, *On the reformulation of Nonlinear Complementarity Problems using the Fischer-Burmeister function*, Applied Mathematics Letters, 12 pp. 7-12, 1999.
- [4] R. Andreani e J. M. Martínez, *On the Solution of Bounded and Unbounded Mixed Complementarity Problems*, Optimization 50, pp. 265-278, 2001.
- [5] R. Andreani e J. M. Martínez, *On the solution of mathematical programming problems with equilibrium constraints*. Mathematical Methods of Operations Research 54, pp. 345-358, 2001.
- [6] R. Andreani e A. Friedlander, *Bound Constrained Smooth Optimization for Solving Variational Inequalities and Related Problems*, Annals of Operations Research 116 (1-4), pp. 179-198, 2002.

- [7] R. Arnott e K. Small, *The economics of traffic congestion*, Boston College Working Papers in Economics 256, Boston College, Department of Economics, 1994.
- [8] J. F. Bard e J.E. Falk, *An Explicit Solution to the Multi-Level Programming Problem*, Computers and Operations Research 9, 77-100, 1982.
- [9] J. F. Bard, *Practical Bilevel Optimization: Algorithms and applications*, Kluwer Academic Publishers, 1998.
- [10] M. S. Bazarra, H. D. Sherali e C. M. Shetty, *Nonlinear Programming : Theory and Algorithms*, Second Edition, John Wiley & Sons, New York, New York, 1993.
- [11] J. F. Bonnans e A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer Series in Operations Research, Springer, 2000.
- [12] J. Bracken e J. T. McGill, *Mathematical programs with optimization problems in the constraints*, Operations Research 21, pp. 37-44, 1973.
- [13] L. Brotcorne, M. Labbé, P. Marcotte e G. Savard, *A Bilevel Model for Toll Optimization on a Multicommodity Transportation Network*, Transportation Science 35-4, pp. 345-358, 2001.
- [14] P. H. Calamai e L. N. Vicente, *Generating Linear and Linear Quadratic Bilevel Programming Problems*, SIAM Journal on Scientific Computing, 14, pp. 770-782, 1993.
- [15] P. H. Calamai e L. N. Vicente, *Generating quadratic bilevel programming test problems*, ACM Transactions on Mathematical Software, 20, pp. 103-119, 1994.
- [16] M. Campêlo, *Programação Linear em Dois Níveis: Uma Abordagem Teórica e Computacional*, Tese de Doutorado, Universidade Federal do Rio de Janeiro, 1999.
- [17] W. Candler e R. Norton, *Multi-Level Programming and Development Policy*, Working Paper 258, World Bank, Washington DC, May 1977.

- [18] S. L. C. Castro, *Técnicas de Restauração Inexata Aplicada à Resolução de Problemas de Programação Matemática em Dois Níveis*, Tese de Doutorado, Departamento de Matemática Aplicada -UNICAMP, Abril 2004.
- [19] Y. Censor, A. N. Iusem e S. Zenios, *An Interior Point Method with Bregman Functions for the Variational Inequality Problem with Paramonotone Operators*, Mathematical Programming, 81, pp. 373-400, 2000.
- [20] J. L. Chela, *Estratégias para a Resolução do Problema de Inequações Variacionais*, Dissertação de Mestrado, Universidade Estadual de São Paulo- Campus de São José do Rio Preto-SP, 02/2002.
- [21] B. Chen, X. Chen and C. Kanzow, *A penalized Fischer Burmeister NCP-Function Theoretical Investigation and Numerical Results*, Mathematical Programming, 88, pp. 211-216, 2000.
- [22] P. A. Clark e A. W. Westerberg, *Bilevel Programming for Steady-State Chemical Process Design-I. Fundamentals and Algorithms*, Computers Chemical Engineering, 14-1, pp. 87-97, 1990.
- [23] B. Colson, *Bilevel Programming with Approximation methods: Software guide and test problems*, Technical Report CRT-38, Centre de Recherche sur les Transports, Université de Montréal, Montréal, Canada, 2002.
- [24] B. Colson, P. Marcotte e G. Savard, *A Trust-Region Method for Nonlinear Bilevel Programming: Algorithm and Computational Experience*, Computational Optimization and Applications, 30, pp. 211-227, 2005.
- [25] S. Dempe, *Foundations of Bilevel Programming*, Kluwer Academic Publishers, Netherlands, 2002.
- [26] S. Dempe, *Annotated Bibliography on Bilevel Programming and Mathematical Problems with equilibrium Constraints*, Optimization 52, pp. 333-359, 2003.

- [27] S. Dempe, *Bilevel Programming - A survey*, Technical Report TU 2003-11, Bergakademie Freiberg, 2003.
- [28] J. E. Falk e J. Liu, *On Bilevel Programming, Part I: General Nonlinear Cases*, Mathematical Programming 70, pp. 47-72, 1995.
- [29] F. Facchinei e J.S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems* Vol. 1 e 2, Springer-Verlag, New York, 2003.
- [30] F. Facchinei, H. Jiang e L. Qi *A Smoothing Method for Mathematical Programs with equilibrium constraint*, Mathematical Programming 85, pp. 107-134, 1999.
- [31] F. Facchinei, A. Fischer e C. Kanzow, *A semi smooth Newton method for variational inequalities :The case of box constraints*, eds. In M.C. Ferris and J.S. Pang (eds), Complementarity and Variational Problems: State of the Art, SIAM, Philadelphia, pp. 76-90, 1997.
- [32] P. Ferrari, *Road network toll pricing and social welfare* Trans. Res. B 36 (5), pp. 471-483, 2002.
- [33] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, Inc., 1983.
- [34] M. Fukushima, *A Relaxed Projection Method for Variational Inequalities*, Mathematical Programming, 35 pp. 58-70 , 1986.
- [35] P. T. Harker e J.S. Pang, *Finite-dimensional variational inequality and nonlinear complementary problems: a survey of theory, algorithms and applications*, Mathematical Programming 48, pp. 161-220, 1990.
- [36] P. T. Harker e J.S. Pang, *On the existence of optimal solutions to mathematical program with equilibrium constraints*, Operations Research Letters 7, pp. 61-64, 1988.

- [37] D. W. Hearn, *Bounding Flows in Traffic Assignment Models*, Research report N.80-4, Dept. of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, 1980.
- [38] D. W. Hearn e M. V. Ramana, *Solving congestion toll pricing models*, In *Equilibrium and Advanced Transportation Modelling*, P. Marcotte, S. Nguyen (eds), Kluwer Academic Publisher, Boston, The Netherlands, pp. 109-124, 1998.
- [39] D. W. Hearn e M. B. Yildirim, *A toll pricing framework for traffic assignment problems with elastic demands*, In: *Current Trends in Transportation and Network Analysis: Miscellanea in Honor of Michael Florian*, M. Gendreau, P. Marcotte(eds), Kluwer Academic Publisher, Dordrecht, The Netherlands, 2001.
- [40] D. W. Hearn e S. Lawphongpanich, *On the Second-Best Toll Pricing Problems*, *Mathematical Programming Series B*, 101, pp. 33-55, 2004.
- [41] D. W. Hearn, P. Bergendorff e M. V. Ramana, *Congestion Toll Pricing of Traffic Networks*, *Network Optimization*, P. M. Pardalos, D.W. Hearn and W.W. Hager (Eds.), *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Vol. 450, pp. 51-71, 1997.
- [42] A. N. Iusem e B. F. Svaiter, *A variant of Korpelevich's method for variational inequalities with a new search strategy*, *Optimization* 42, pp. 309-321, 1997.
- [43] A. F. Izmailov, *Mathematical Programs with Complementarity Constraints: Regularity, Optimality Conditions, and Sensitivity*, *Comp. Mathematics and Mathematical Physics*, 44-7, pp. 1145-1164, 2004.
- [44] O. Johansson-Stenman e T. Sterner, *What is the scope for environmental road pricing?* *Road pricing Traffic Congestion and Environment*, K.J. Button, E.T. Verhoef (eds.), Edward Elgar Publishing Limited, London, England, 1998.

- [45] M. Labbé, P. Marcotte e G. Savard, *A bilevel model of taxation and its application to optimal highway pricing*, Manage. Sci, 44-12, pp. 1608-1622, 1998.
- [46] Z. Q. Luo, J. S. Pang e D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, 1996.
- [47] P. Marcotte e D. Zhu, *Exact and Inexact Penalty Methods for the Generalized Bilevel Programming Problem*, Mathematical Programming, 74, 141-157, 1996.
- [48] P. Marcotte, *A new algorithm for solving variational inequalities with application to the traffic assignment problem*, Mathematical Programming, 33, pp. 339-351, 1985.
- [49] P. Marcotte, *Network design problem with congestion effects*, Mathematical Programming 34, 142-162 1986.
- [50] J. M. Martínez, *Inexact-Restoration Method with Lagrangian Tangent Decrease and New Merit Function for Nonlinear Programming*, JOTA, 111-1, pp. 39-58, 2001.
- [51] J. M. Martínez, *Two-Phase Model Algorithm with Global Convergence for Nonlinear Programming*, JOTA, 96(2), 397-436, 1998.
- [52] J. M. Martínez e E.A. Pilotta, *Inexact-Restoration Algorithm for Constrained Optimization*, JOTA, 104(1), 135-163, 2000.
- [53] J. M. Martínez e E.A. Pilotta, *Inexact Restoration Methods for Nonlinear Programming: Advances and Perspectives*, Optimization and Control with applications, edited by L.Q.Qi, K.L.Teo and X.Q.Yang., Kluwer Academic Publishers, 2003.
- [54] J. M. Martínez e B.F. Svaiter, *A Practical Optimality Condition without Constraint Qualifications for Nonlinear Programming*, JOTA, 118, 117-133, 2003.
- [55] A. Migdalas, *Bilevel Programming in traffic planning: models, methods and challenge*, Journal of Global Optimization, 4 (2): 340-357, 1994.

- [56] J. V. Outrata, *Necessary Optimality Conditions for Stackelberg Problems*, JOTA, 76, 305-320, 1993.
- [57] J. V. Outrata, *On Optimization Problems with Variational Inequality Constraints*, SIAM Journal Optimization, 4, 340-357, 1994.
- [58] J. V. Outrata e J. Zowe, *A Numerical Approach to Optimization Problems with Variational Inequality Constraints*, Mathematical Programming, 68, 105-130, 1995.
- [59] M. Patriksson e R. T. Rockafellar, *A Mathematical model and descent algorithm for bilevel traffic management*, Trans. Sci 36 (3), 271-291, 2002.
- [60] R. T. Rockafellar, *Convex Analysis*, Princeton, New Jersey-Princeton University Press, 1970.
- [61] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill Inc., 1976.
- [62] S. Scheimberg e M. Campelo, *A study of local solutions in linear bilevel programming*, Journal of Optimization Theory and Applications, 125-1, pp. 63-84, 2005.
- [63] S. Scheimberg., M. Campelo e C. H. Saboia, *A computational study of global algorithms for linear bilevel programming*, Numerical Algorithms, 35, pp. 155-173, 2004.
- [64] S. Scheimberg e M. Campelo, *Theoretical and Computational Results for a Linear Bilevel Problem*, Nonconvex Optimization and its Applications, 54, pp. 269-281, 2001.
- [65] K. Shimizu, Y. Ishizuka e J.F. Bard, *Nondifferentiable and Two-Level Mathematical Programming*, Kluwer Academic Press, Boston, 1997.
- [66] M. V. Solodov e B. F. Svaiter, *A New Projection Method for Variational Inequality Problems*, SIAM Journal Control Optimization, 37 - 3, pp. 765-776, 1999.

- [67] L. N. Vicente, *Programação de Dois Níveis*, Tese de Mestrado, Departamento de Matemática, Universidade de Coimbra, 1992.
- [68] L. N. Vicente e P.H. Calamai, *Geometry and Local Optimality Conditions for Bilevel Programs with Quadratic Strictly Convex Lower Levels*, Nonconvex Optimization and its Applications, 4, 141–151, Kluwer Academic Publishers, (Report available at FTP: dial.uwaterloo.ca), 1995.
- [69] L.N. Vicente, *Bilevel Programming: Introduction, History and Overview*, Encyclopedia of Optimization, ed. by C. A. Floudas and P. M. Pardalos, vol. 1, pp. 178-180, Kluwer Academic Publishers, Dordrecht, 2001
- [70] L.N. Vicente e P.H. Calamai, *Bilevel and Multilevel Programming. A Bibliography Review*, Journal of Global Optimization, 5, 291-306. 1994.
- [71] Y. Wang, Y.C. Jiao e H. Li, *An Evolutionary Algorithm for Solving Nonlinear Bilevel Programming Based on a New Constraint-Handling Scheme* IEEE Transactions on Systems, Man, and Cybernetics- Part C: Applications and Reviews, vol. 35, n.2 May 2005.
- [72] F.S. Yano, *Estratégias para a Resolução do Problema MPEC*, Dissertação de Mestrado, Universidade Estadual de São Paulo- Campus de São José do Rio Preto- SP, 02/2003.
- [73] J. J. Ye, *Optimality Conditions for Optimization Problems with Complementarity Constraints*, SIAM Journal Optimization, 9-2, pp. 374-387, 1999.