

Laboratório Nacional de Computação Científica
Programa de Pós Graduação em Modelagem Computacional

**A Evolução Diferencial e o Algoritmo de Lemke na
Resolução de Problemas de Otimização em Dois Níveis**

Por

Celio Henrique Nogueira Larcher Jr.

PETRÓPOLIS, RJ - BRASIL

FEVEREIRO DE 2017

**A EVOLUÇÃO DIFERENCIAL E O ALGORITMO DE LEMKE NA
RESOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO EM DOIS
NÍVEIS**

Celio Henrique Nogueira Larcher Jr.

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO LABORATÓRIO
NACIONAL DE COMPUTAÇÃO CIENTÍFICA COMO PARTE DOS REQUISI-
TOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
CIÊNCIAS EM MODELAGEM COMPUTACIONAL

Aprovada por:

Prof. Helio José Corrêa Barbosa, D.Sc.
(Presidente)

Prof. Paulo Cesar Marques Vieira, D.Sc.

Prof. Fernanda Maria Pereira Raupp, D.Sc.

Prof. Heder Soares Bernardino, D.Sc.

PETRÓPOLIS, RJ - BRASIL
FEVEREIRO DE 2017

Larcher Jr., Celio Henrique Nogueira

L319e a evolução diferencial e o algoritmo de lemke na resolução de problemas de otimização em dois níveis / Celio Henrique Nogueira Larcher Jr.. Petrópolis, RJ. : Laboratório Nacional de Computação Científica, 2017.

CXV, 115 p. : il.; 29 cm

Orientador: Helio José Corrêa Barbosa

Dissertação (M.Sc.) – Laboratório Nacional de Computação Científica, 2017.

1. Otimização estrutural 2. Evolução diferencial 3. Otimização em dois níveis I. Barbosa, Helio José Corrêa. II. LNCC/MCT. III. Título.

CDD – 003

”O Universo, como já foi dito anteriormente, é um lugar desconcertantemente grande, um fato que, para continuar levando uma vida tranquila, a maioria das pessoas tende a ignorar.”

Douglas Adams: O guia do mochileiro das galáxias

À meus pais e irmãos,
minha companheira e amigos.

Agradecimentos

Primeiramente gostaria de agradecer a minha família, meus pais Celio e Helena e meus irmãos Lucas e Carolina, por todo apoio e carinho que sempre demonstraram.

Não poderia esquecer minha namorada Viviane que tanto me acompanhou durante o mestrado, tornando mais leve e fácil esta jornada.

Ainda, a meus velhos amigos de Juiz de Fora, especialmente Marcio e João Marcos, que mesmo com uma pequena dificuldade logística não se perdeu o contato e, muito menos, a amizade.

Aos professores e colegas do LNCC que colaboraram com minha formação por todo este período.

Ao meu orientador Helio Barbosa por me guiar e auxiliar durante este processo, sem o qual tudo teria sido absurdamente mais difícil.

E a todos os outros que, de alguma forma, contribuíram durante esta longa e intensa caminhada.

Resumo da Dissertação apresentada ao LNCC/MCT como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

A EVOLUÇÃO DIFERENCIAL E O ALGORITMO DE LEMKE NA RESOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO EM DOIS NÍVEIS

Celio Henrique Nogueira Larcher Jr.

Fevereiro , 2017

Orientador: Helio José Corrêa Barbosa, D.Sc.

O estudo do problema de otimização em dois níveis desperta particular interesse dada sua capacidade de representar estruturas hierárquicas de decisão, mas as dificuldades na sua resolução exigem o desenvolvimento de técnicas computacionais adequadas. Este trabalho tem como foco a solução de uma classe de problemas em dois níveis onde a função objetivo do agente seguidor é linear ou quadrática nas variáveis do seguidor e com restrições lineares.

Utilizando-se então as condições de Karush-Kuhn-Tucker, verificou-se a possibilidade de resolução do nível do seguidor como um problema de complementaridade linear, dando origem a uma abordagem aninhada. Nesta, a resposta do seguidor é obtida pelo algoritmo de Lemke, enquanto o nível do líder é tratado pela metaheurística evolução diferencial, tendo em vista os bons resultados desta e sua grande generalidade.

A técnica proposta é submetida a um conjunto de testes utilizando instâncias da literatura bem como outras, propostas neste trabalho, com o objetivo de validar a utilização do algoritmo de Lemke e estudar seu desempenho computacional.

Abstract of Dissertation presented to LNCC/MCT as a partial fulfillment of the requirements for the degree of Master of Sciences (M.Sc.)

**THE DIFFERENTIAL EVOLUTION AND THE LEMKE
ALGORITHM IN THE SOLVE OF BILEVEL PROGRAMMING
OPTIMIZATION PROBLEMS**

Celio Henrique Nogueira Larcher Jr.

February, 2017

Advisor: Helio José Corrêa Barbosa, D.Sc.

The study of bilevel optimization problems arouses particular interest given their ability to represent hierarchical decision structures, but the challenges in their resolution require the development of appropriate computational techniques. This work aims at the solution of a class of bilevel problems where the objective function in the follower agent is linear or quadratic in the follower variables and linear constraints may be present.

With the use of the Karush-Kuhn-Tucker conditions, it was verified the possibility of solving the follower level as a linear complementarity problem, giving rise to a nested approach. In this approach, the follower response is obtained by Lemke's algorithm, while the leader level is treated with the differential evolution metaheuristic, in view of its good results and great generality.

The proposed technique is submitted to a test-set using instances from the literature, as well as others, proposed in this work, with the intent of validating the use of Lemke's algorithm in the follower level, and study its computational performance.

Sumário

1	Introdução	1
1.1	O problema de programação em dois níveis	2
1.2	Utilização de metaheurísticas	3
1.3	Objetivos do trabalho	4
2	Fundamentação Teórica	7
2.1	Condições de Karush–Kuhn–Tucker	7
2.2	Problema de Complementaridade Linear	9
2.2.1	Algoritmo de Lemke	12
2.2.2	Exemplos de execução para o algoritmo de Lemke	16
2.2.3	Convergência do Algoritmo de Lemke	20
2.3	O Problema de Programação em Dois Níveis	21
2.3.1	Definição Matemática	24
2.3.2	Análise de otimalidade no BLPP	27
2.3.3	O BLPP como um problema de complementaridade linear	28
2.3.4	Aplicação de Metaheurísticas ao BLPP	32
3	Evolução Diferencial	35
3.1	Fluxo Principal	36
3.2	Inicialização	39
3.3	Mutação	39
3.4	Recombinação	42

3.5	Avaliação	43
3.6	Tratamento de Restrições	44
4	Algoritmo DE+Lemke	46
4.1	Estrutura	46
4.2	Otimização no nível do líder via DE	48
4.2.1	Método de Deb aplicado ao DE+LEMKE	49
4.3	Resolução via Lemke para o seguidor	50
5	Casos de Teste	51
5.1	Instâncias-Teste na Literatura	51
5.2	Instâncias-Teste Escaláveis	60
5.2.1	Instâncias-Teste sem Restrições Funcionais	61
5.2.2	Instâncias-Teste com Restrições Funcionais	68
6	Experimentos Computacionais	75
6.1	Ajuste de parâmetros	76
6.2	Análise de custo computacional do algoritmo de Lemke	76
6.3	Comparação da técnica DE+LEMKE a um algoritmo DE+DE . . .	85
6.3.1	Instâncias da literatura	86
6.3.2	Instâncias escaláveis	91
6.4	Desempenho global da técnica DE+LEMKE	95
7	Conclusão e Trabalhos Futuros	107
	Referências Bibliográficas	110

Lista de Figuras

Figura

2.1	Exemplo: cone viável para o LCP	11
2.2	Exemplo: cone inviável para o LCP	11
5.1	Função objetivo do Nível do Líder: SMDquad1	62
5.2	Função objetivo do Nível do Líder: SMDquad2	64
5.3	Função objetivo do Nível do Líder: SMDquad3	65
5.4	Função objetivo do Nível do Líder: SMDquad4	66
5.5	Função objetivo do Nível do Líder: SMDquad5	67
5.6	Função objetivo do Nível do Líder: Cquad1.	70
5.7	Função objetivo do Nível do Líder: Cquad2.	71
5.8	Função objetivo do Nível do Líder: Cquad3.	72
5.9	Função objetivo do Nível do Líder: Cquad4.	74
6.1	Literatura - Operações de pivotamento x Dimensão x Restrições (Líder)	78
6.2	Literatura - Operações de pivotamento x Dimensão x Restrições (Seguidor)	79
6.3	SMDquad - Operações de pivotamento x Dimensão (Lider/Seguidor)	81
6.4	Cquad - Operações de pivotamento x Dimensão (Lider/Seguidor) x Restrições (Lider/Seguidor)	82
6.5	Literatura - Operações de pivotamento x Dimensão matriz de Lemke	83
6.6	SMDquad - Operações de pivotamento x Dimensão matriz de Lemke	84
6.7	Cquad - Operações de pivotamento x Dimensão matriz de Lemke .	85

6.8	Perfil de Desempenho das técnicas DE+LEMKE, R-DE+LEMKE e DE+DE em instâncias escaláveis	105
6.9	Perfil de Desempenho das técnicas DE+LEMKE, R-DE+LEMKE e DE+DE em instâncias escaláveis ($\tau \in [1, 2]$)	106

Lista de Tabelas

Tabela

5.1	Casos de Teste da Literatura	59
5.2	SMD1 x SMDquad1	62
5.3	SMD2 x SMDquad2	63
5.4	SMD3 x SMDquad3	64
5.5	SMD4 x SMDquad4	65
5.6	SMD5 x SMDquad5	67
5.7	Descrição da instância Cquad1.	69
5.8	Descrição da instância Cquad2.	70
5.9	Descrição da instância Cquad3.	72
5.10	Descrição da instância Cquad4.	73
6.1	Operações de pivotamento e característica dos problemas - Instâncias da literatura	77
6.2	Operações de pivotamento e característica dos problemas - Instâncias escaláveis	80
6.3	Comparação DE x Lemke para instâncias da literatura	87
6.4	Comparação DE+LEMKE x DE+DE para instâncias da literatura .	90
6.5	Comparação DE x Lemke para instâncias escaláveis	92
6.6	Comparação DE+LEMKE x DE+DE para instâncias escaláveis . .	93
6.7	Comparação das melhores solução obtidas nas técnicas DE+LEMKE, R-DE+LEMKE e DE+DE em instâncias da literatura. O superscrito “a” indica um problema de maximização em ambos os níveis. .	98

6.8	Comparação dos resultados obtidos pelas técnicas DE+LEMKE, R-DE+LEMKE e DE+DE em instâncias da literatura. O superscrito “a” indica um problema de maximização em ambos os níveis.	100
6.9	Comparação dos resultados obtidos pelas técnicas DE+LEMKE, R-DE+LEMKE e DE+DE em instâncias escaláveis	104

Lista de Siglas e Abreviaturas

- BLPP: Problema de programação em dois níveis (Bilevel Programming Problem).
- KKT: Condições de Karush-Kuhn-Tucker
- LCP: Problema de complementaridade linear (Linear Complementarity Problem).
- DE: Evolução Diferencial (Differential Evolution)
- MPEC: Problemas de programação matemática com restrições de equilíbrio (Mathematical programming with equilibrium constraints)

Capítulo 1

Introdução

O conceito de otimização é intrínseco à ideia de se fazer mais com menos, aproveitando os recursos disponíveis da melhor maneira possível. Dado seu caráter universal, pode ser considerada inclusive uma necessidade natural, organismos com melhor aproveitamento energético tendem a permanecer em populações por mais tempo, da mesma forma que moléculas buscam assumir configurações de menor nível energético.

Do ponto de vista matemático, problemas de otimização estão vinculados a uma função matemática, denominada função objetivo, que é capaz de avaliar cada ponto do espaço de busca e quantificar a escolha deste ponto através de uma pontuação. O processo de otimização se dá pela exploração do espaço de busca, guiada por esta função objetivo, visando encontrar o ponto de melhor avaliação possível.

Além da função objetivo, outro componente comum à formulação de um problema de otimização é a estipulação de restrições, sendo estas responsáveis por impedir que o processo exploratório atinja determinadas regiões do espaço não desejáveis. A estipulação de múltiplos objetivos pode também ser citada dada a abordagem distinta que exige do problema, impondo uma busca em relação a dois ou mais aspectos de interesse não necessariamente relacionados e mesmo incomensuráveis e conflitantes.

Este trabalho trata de outra variante presente em problemas de otimização,

a programação em dois níveis. Nesta, além do objetivo comum a problemas de otimização, é estipulado um segundo objetivo a ser tratado em nível de restrição. A formulação resultante do acréscimo deste segundo objetivo induz uma estrutura hierárquica, com objetivos preferências a serem almejados e objetivos secundários a serem atendidos durante a busca, o que trás comportamento bem peculiar a este problema e o distingue do problema de otimização multiobjetivo.

1.1 O problema de programação em dois níveis

Um assunto de crescente interesse no contexto acadêmico, a programação em dois níveis vem aparecendo como boa alternativa para representar diversos casos de interação entre agentes.

Como comentado anteriormente, o problema de programação em dois níveis pode ser descrito como um problema de otimização que possui, entre suas restrições, outro problema de otimização a ser resolvido. Neste contexto, a função externa é denominada líder e representa o papel de um agente tomador de decisões capaz de determinar suas escolhas a revelia de qualquer outro personagem, enquanto o problema presente em nível de restrições se comporta como um agente seguidor sendo “influenciável” pelo líder e tomando suas decisões baseadas nas escolhas do primeiro agente. Ainda, apesar de decisões do líder serem tomadas a revelia de seu seguidor, este não é isolado da resposta do agente que o segue, tendo o comportamento do seguidor efeitos sobre os interesses do líder.

Outro fator digno de nota e tão importante quanto a prioridade na tomada de decisão, o líder deste processo é capaz de antever a reação de outros agentes, podendo antecipar o cenário futuro para suas escolhas e determinar a melhor decisão a ser tomada, não somente por seus próprios interesses, mas pelas reações dos demais.

Exemplos onde esta estrutura de tomada de decisões hierárquica pode ser encontrada aparecem em diversos contextos podendo ser citados aplicações militares em planejamento defensivo frente a ataques (Bracken e McGill, 1974), alocação de

instalações de facilidades considerando ações de concorrência (Beresnev, 2013), estabelecimento de estratégia ótima de preços frente a concorrentes (Labbé e Violin, 2013), arquitetura de redes de diversas naturezas (Constantin e Florian, 1995; Costantino et al., 2013), estabelecimento de políticas de controle e incentivo a perfis de geração de energia (Bard et al., 2000), entre diversos outros.

Apesar de ser um problema útil do ponto de vista teórico e prático, o problema de programação em dois níveis possui uma grande dificuldade no que tange ao desenvolvimento de métodos eficientes para sua resolução. A caracterização matemática como um problema de otimização que contém, entre suas restrições, outro problema de otimização a ser solucionado traz enormes dificuldades no desenvolvimento de métodos, dificuldades que são acentuadas ao se considerar que não é possível, na maior parte dos casos, garantir diferenciabilidade ou mesmo convexidade. De fato é um problema NP-Difícil (Hansen et al., 1992).

Considerando a complexidade nesta formulação, é comum em diversos trabalhos a manipulação deste problema de forma a que este assuma aspecto mais amigável, especialmente eliminando a otimização no seguidor. Uma possibilidade neste contexto é a reformulação como um problema de otimização com restrições de equilíbrio através das condições de Karush-Kuhn-Tucker, abordagem que não é sempre possível e traz outras dificuldades referentes às restrições de equilíbrio adicionadas no processo.

Em outra linha, a própria garantia de otimalidade vem sendo preterida pela utilização de métodos competitivos, mesmo que apresentem resultados aproximados e sem garantia. Nesta linha, vem-se observando um crescente movimento voltado ao uso de métodos estocásticos, ponto em que a utilização de metaheurística é uma opção de destaque.

1.2 Utilização de metaheurísticas

Uma questão recorrente na busca pela resolução de problemas em um contexto computacional é a dificuldade em se encontrar uma sequência eficiente de

comandos capaz de realizar determinada tarefa. Algoritmos determinísticos são, muitas vezes, demasiadamente custosos o que torna sua execução proibitiva.

Neste contexto, a utilização de métodos heurísticos ou estocásticos pode ser uma alternativa válida dado que permite buscar por boas configurações do espaço de busca sem necessidade de vasculhar a totalidade do mesmo.

Na classe dos métodos heurísticos, as metaheurísticas são técnicas de propósito geral que visam, através de um conjunto de operações, vasculhar de forma sistemática o espaço de busca de interesse à procura da melhor configuração que atenda determinada função de avaliação. Sua grande vantagem está em possuir uma formulação genérica que permite ser implementada nos mais diversos contextos, além de apresentar, em geral, boa eficácia nestas implementações.

Outro ponto atraente em metaheurísticas são as poucas exigências que fazem acerca de suas implementações. Isto permite a inclusão de uma gama de outros recursos que podem ser atrelados à execução dessas metaheurísticas, aprimorando o desempenho das mesmas ou delegando parte do trabalho necessário nesta execução.

Neste contexto ainda, *matheuristics* tem se apresentado como uma boa alternativa dada a grande variedade de métodos matemáticos eficientes disponíveis para problemas “bem comportados”, mas que não são capazes de resolver outros problemas que demandam grande esforço computacional. A implementação de uma abordagem como esta visa a derivação de um problema complexo em subproblemas que possuam métodos matemáticos a serem explorados, mantendo aqueles trechos da formulação em que isto não é possível, ou não é desejável, a cargo de uma metaheurística.

1.3 Objetivos do trabalho

O trabalho aqui proposto tem como foco uma classe específica da programação em dois níveis, quando tem-se o problema do seguidor representado por uma função objetivo linear ou quadrática e restrições exclusivamente lineares, sendo estas características relativas apenas às variáveis do próprio seguidor.

Através desta especificação permite-se levantar alguns pressupostos acerca do problema, como a possibilidade de se aplicar as condições de Karush-Kuhn-Tucker, viabilizando a utilização de métodos promissores para resolução do problema no nível do seguidor.

Neste contexto é apresentado um método de resolução através do algoritmo de Lemke de forma a se obter, para cada configuração do líder, a resposta exata (o termo exato neste trabalho é utilizado desconsiderando erros decorrentes da aritmética finita dos computadores). Apesar de sua popularidade na resolução de problemas de complementaridade linear, a utilização do método de Lemke se encontra pouco explorada em sua associação com o problema de programação em dois níveis, tendo sido observado apenas um trabalho fazendo uso do mesmo procedimento aqui empregado (Li e Wang, 2006).

É então proposta a técnica DE+LEMKE, uma *matheuristic* que, utilizando-se de resolução aninhada, computa a resposta do seguidor via método de Lemke, enquanto para o nível do líder realiza a busca através da metaheurística evolução diferencial. A utilização da evolução diferencial é desejável dado que não exige a imposição de pressupostos ao problema (característica das metaheurísticas) e apresenta bons resultados na literatura, inclusive quando aplicada ao próprio BLPP (Angelo et al., 2013).

Ainda, para uma melhor análise da técnica e tendo em conta a pouca opção de instâncias que apresentem desafio à abordagem (considerando o subproblema tratado), foi proposto um novo conjunto de instâncias teste escaláveis que permitem uma melhor verificação de desempenho.

Tem-se então a avaliação de desempenho do método de Lemke, tanto relacionado ao número de passos necessários para convergência do algoritmo, quanto quando comparado a uma implementação do DE no nível do seguidor, de forma a tentar verificar o custo da técnica e possíveis prós e contras em se utilizar uma técnica de resolução exata.

Por fim, é feita uma análise geral de desempenho da técnica DE+LEMKE

comparando-a a outras técnicas desenvolvidas no decorrer deste trabalho.

O trabalho se divide em 7 capítulos: o capítulo corrente (Capítulo 1) apresenta uma breve introdução sobre os temas abordados, ressaltando características de interesse do problema e abordagem utilizada na resolução; o Capítulo 2 apresenta os conceitos necessários para entendimento da proposta, bem como referências acerca de iniciativas correlatas; o Capítulo 3 apresenta uma descrição geral da metaheurística evolução diferencial, com os principais fatores de implementação da mesma e variações comuns na literatura; o Capítulo 4 apresenta a abordagem proposta DE+LEMKE para resolução do problema de programação em dois níveis; o Capítulo 5 apresenta os casos de teste utilizados para validação da técnica, incluindo os novos casos de teste propostos; o Capítulo 6 apresenta os experimentos computacionais realizados, bem como comenta os resultados e conclusões acerca dos mesmos; por fim o Capítulo 7 apresenta um fechamento sobre o trabalho, com as principais conclusões, bem como propostas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo são apresentados os conceitos teóricos básicos envolvidos no desenvolvimento deste trabalho, bem como um panorama geral para o problema de programação em dois níveis.

Desta forma, inicialmente são expostas as premissas básicas de utilização das condições de Karush-Kuhn-Tucker, a definição de um problema de complementaridade linear e o procedimento de aplicação do algoritmo de Lemke, bem como algumas condições de convergência. Segue-se então ao próprio problema de programação em dois níveis, onde tem-se uma descrição do mesmo, trabalhos relacionados, modelo matemático da classe específica tratada neste trabalho e a reescrita do nível do seguidor como um problema de complementaridade linear.

2.1 Condições de Karush–Kuhn–Tucker

As condições de Karush-Kuhn-Tucker (abreviadas como KKT) foram apresentadas inicialmente em um trabalho de mestrado de William Karush em 1939 e republicadas posteriormente por Harold Kuhn e Albert Tucker em 1950.

O grande interesse contido neste resultado teórico vem por apresentar as condições de otimalidade dos problemas, com ou sem restrições, através de um conjunto de equações que possuem solução apenas em pontos críticos da função objetivo. Desta maneira, é possível trocar, em alguns casos, o problema de otimização pela investigação dos pontos críticos da função em busca de seu mínimo (nesta

definição é considerada um problema de minimização, sem perda de generalidade).

As condições KKT são apresentadas nos Teoremas 1, 2 e 3.

Teorema 1 Condição de necessária de primeira ordem

Sejam $f(x) : \mathbf{R}^n \rightarrow \mathbf{R}$ uma função objetivo e $g_i(x) : \mathbf{R}^n \rightarrow \mathbf{R}$, $i = 1, \dots, m$ e $h_j(x) : \mathbf{R}^n \rightarrow \mathbf{R}$, $j = 1, \dots, k$ restrições de desigualdade e igualdade, respectivamente. Assumindo estas funções continuamente diferenciáveis em um ponto \bar{x} , se \bar{x} é minimizador local, então existem escalares $\bar{\lambda}_i$ e $\bar{\mu}_j$ que satisfazem as condições de Karush-Kuhn-Tucker (Equação 2.1).

$$\begin{aligned} \nabla_x f(x) + \sum_{i=1}^m \lambda_i \nabla_x g_i(x) + \sum_{j=1}^k \mu_j \nabla_x h_j(x) &= 0 \\ \lambda_i g_i(x) &= 0 \\ g_i(x) \leq 0, h_j(x) &= 0 \\ \lambda_i \geq 0, \mu_j &\in \mathbf{R} \end{aligned} \tag{2.1}$$

Teorema 2 Condição necessária de segunda ordem

Sejam $f(x) : \mathbf{R}^n \rightarrow \mathbf{R}$ uma função objetivo e $g_i(x) : \mathbf{R}^n \rightarrow \mathbf{R}$, $i = 1, \dots, m$ e $h_j(x) : \mathbf{R}^n \rightarrow \mathbf{R}$, $j = 1, \dots, k$ restrições de desigualdade e igualdade, respectivamente. Denota-se o conjunto de restrições ativas em um ponto \bar{x} como $J(\bar{x})$. Assumindo estas funções duas vezes diferenciáveis em um ponto \bar{x} que satisfaz as condições KKT e sendo $\bar{\lambda}_i$ e $\bar{\mu}_j$ os multiplicadores relativos a este ponto, então se \bar{x} é minimizador local e $d \neq 0$, tal que:

$$d \in \left\{ \begin{array}{l} \nabla g_i(\bar{x})^T d = 0, \quad g_i(\bar{x}) \in J(\bar{x}), \bar{\lambda}_i > 0 \\ \nabla g_i(\bar{x})^T d \leq 0, \quad g_i(\bar{x}) \in J(\bar{x}), \bar{\lambda}_i = 0 \\ \nabla h_j(\bar{x})^T d = 0 \end{array} \right\}$$

deve-se verificar para todo vetor d

$$d^T \nabla_{xx}^2 L(\bar{x}, \bar{\lambda}, \bar{\mu}) d \geq 0$$

onde $L(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{j=1}^k \mu_j h_j(x)$ é a função Lagrangiana relacionada a $f(x)$.

Teorema 3 Condição suficiente de segunda ordem

Verificando as mesmas hipóteses presentes no Teorema 2, se para todo vetor d tem-se:

$$d^T \nabla_{xx}^2 L(\bar{x}, \bar{\lambda}, \bar{\mu}) d > 0$$

então \bar{x} é minimizador local estrito do problema de otimização.

Um caso particular ocorre quando a função objetivo $f(x)$ e suas restrições $g(x)$ e $h(x)$ são funções convexas, com as condições necessárias de primeira ordem, tornando-se condições suficientes e \bar{x} sendo minimizador global do problema de minimização.

2.2 Problema de Complementaridade Linear

O Problema de Complementaridade Linear (denotado pela sigla LCP) é uma formulação geral em programação matemática que unifica conceitos de programação linear e quadrática, com propriedades dos jogos bimatrix, aparecendo em contextos diversos como na teoria de jogos, programação em dois níveis, análise de estruturas elástico-plásticas, modelos financeiros, entre outros (Murty, 1997).

Matematicamente pode ser descrito da seguinte forma: dados uma matriz quadrada M de ordem p e um vetor $q \in \mathbf{R}^p$, o LCP busca dois vetores w, z positivos e complementares entre si que satisfaçam a relação $w - Mz = q$. A descrição matemática para o problema pode ser vista na Equação 2.2.

$$\begin{aligned} w - Mz &= q \\ w_i z_i &= 0, \quad i = 1 \dots p \\ w_i &\geq 0, z_i \geq 0, \quad i = 1 \dots p \end{aligned} \tag{2.2}$$

A solução (w, z) que atende a estas condições é denominada uma solução viável complementar. Ainda, caso para cada escolha entre os elementos do par (w_j, z_j) , $j = 1 \dots p$, os vetores correspondentes na matriz expandida $[I \quad -M]$ formarem uma base (onde w_j está relacionado à coluna I_j da matriz identidade

e z_j à coluna $-M_j$ da matriz de entrada do LCP), a solução (w, z) é dita uma solução básica viável complementar. É relevante comentar que em uma solução básica a escolha de (w, z) é única para a base em questão mesmo que esta base não seja única para o LCP.

Outra maneira para se interpretar o problema é considerando seu caráter geométrico. Dado dois conjuntos de vetores ordenados, um composto pelas colunas da matriz identidade de dimensão p (base canônica ordenada) e outro conjunto composto pelos elementos das colunas da matriz $-M$, é possível formar cones através da escolha de um entre cada par destes vetores ordenados, sendo cada um destes cones denominado cone complementar. Busca-se um dentre os cones complementares que contenha o vetor q , sendo este o cone complementar associado ao LCP. Nota-se que existem 2^p cones complementares possíveis, ponto onde pode-se perceber o crescimento do espaço de busca ao se aumentar a dimensão.

Existem ainda dois pontos dignos de nota. Um destes é que a determinação de um conjunto de vetores pode levar a um cone degenerado (quando os vetores selecionados são linearmente dependentes entre si), tendo este cone interior vazio mas não inviabilizando a existência de uma solução, *a priori*, e sendo esta possível solução não-básica. Outra situação possível é que, para uma dada matriz M , não exista nenhum cone dentre os possíveis de serem formados pelo LCP que contenha q , não existindo então uma solução viável.

A Figura 2.1 apresenta o exemplo de um LCP com solução básica complementar viável, enquanto a Figura 2.2 tem como exemplo um LCP onde não é possível se obter uma solução.

Neste primeiro caso (Figura 2.1), o LCP é representado pela matriz $M = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$, com os possíveis cones representados pelos pares (I_1, I_2) , $(I_1, -M_2)$, $(I_2, -M_1)$ e $(-M_1, -M_2)$, conforme a condição de complementaridade. É possível observar que, como todos os pares de vetores possíveis são linearmente independentes, todos os cones possuem interior não-vazio. Tendo ainda o vetor $q = [1 \ -1]^T$, uma solução para o LCP em questão seria $w = [2 \ 0]^T$ e $z = [0 \ 1]^T$ relativa ao

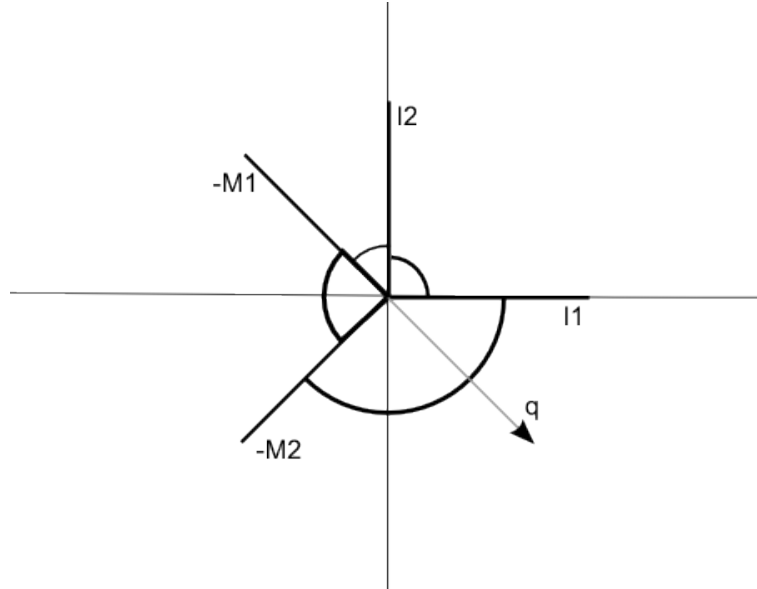


Figura 2.1: Exemplo: cone viável para o LCP

cone formado pelo par $(I_1, -M_2)$, que atende a todas as restrições do problema.

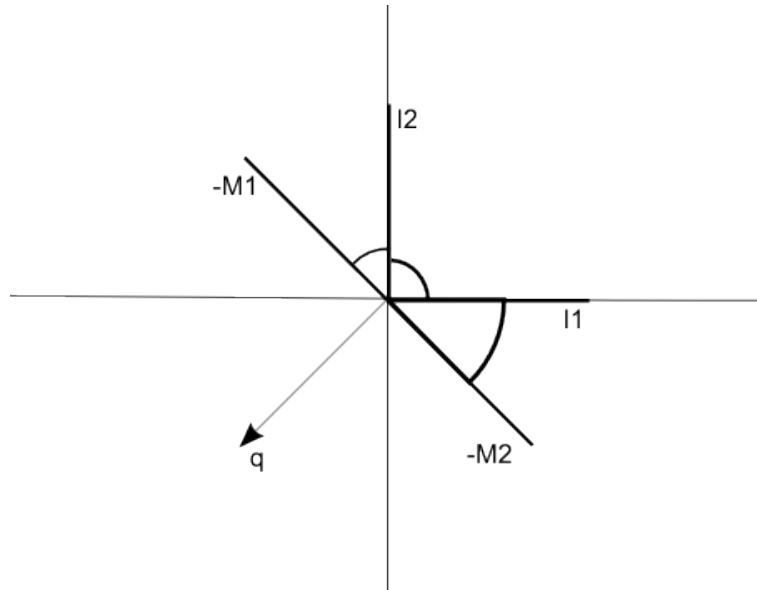


Figura 2.2: Exemplo: cone inviável para o LCP

No segundo caso (Figura 2.2) tem-se o LCP representado pela matriz $M = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$, com os possíveis cones representados pelos pares (I_1, I_2) , $(I_1, -M_2)$, $(I_2, -M_1)$ e $(-M_1, -M_2)$, conforme a condição de complementaridade. Neste, existem vetores linearmente dependentes entre si (vetores $-M_1$ e $-M_2$), fazendo

com que um dos cones possíveis seja degenerado, tendo o cone gerado por estes dois vetores interior vazio. É possível ainda notar graficamente que o vetor $q = [-1 \ -1]^T$ não está incluso em nenhum cone viável, fazendo com que este LCP não possua solução. Repare que não é o cone degenerado que impede que o LCP possua solução (caso o vetor em questão fosse $q = [1 \ -1]^T$ haveria solução para o LCP, podendo inclusive ser o próprio cone degenerado mas com infinitos valores possíveis para (w, z)), apenas torna determinadas combinações de cones pouco úteis.

Em uma primeira análise o método mais prático para resolução do LCP é através da enumeração de todos os cones complementares possíveis, verificando se cada escolha é solução para o problema via resolução de um sistema linear. Apesar deste método ser conceitualmente simples, esta é uma opção proibitiva do ponto de vista computacional dado que, como comentado anteriormente, há 2^p cones complementares possíveis a serem analisados.

Desta forma, outros algoritmos foram propostos na literatura para resolução do problema de complementaridade. Neste trabalho foi escolhido o Algoritmo de Lemke, sendo um dos mais populares na literatura (Massaro e Pelillo, 2001) e com garantias de convergência para um bom número de casos.

2.2.1 Algoritmo de Lemke

Proposto por Carlton Lemke (Lemke, 1965), o Algoritmo de Lemke resumidamente busca, através de sucessivas operações de pivotamento, encontrar um par de vetores (w, z) que sejam solução para o LCP, dados uma matriz M e um vetor q . Para isso, busca estabelecer uma base inicial viável com auxílio de uma variável artificial e realizar sucessivas trocas de elemento nesta base até que a variável artificial seja removida.

Para compreender o funcionamento do algoritmo é necessário recorrer ao conceito de bases vinculado à ideia de sistemas subdeterminados. Considerando uma matriz A de dimensão $m \times n$, $m \leq n$, e posto m em um sistema da forma $Ax = b$, uma base B para o sistema equivale a uma matriz quadrada de ordem

m , não-singular, formada pela escolha de m colunas da matriz A . Utilizando-se desta divisão e considerando D o conjunto de colunas não escolhidas para a base, o sistema poderia ser reescrito como:

$$Bx_B + Dx_D = b$$

sendo x_B considerado o conjunto de variáveis associadas à base (variáveis básicas) e x_D o conjunto de variáveis fora da base (variáveis não-básicas). Nesta configuração, uma maneira simples de resolução do problema é considerar $x_D = 0$ e $x_B = B^{-1}b$.

Uma maneira comum de se referir a este tipo de formulação é através do formato *tableau*, uma forma de escrita que facilita a execução de algoritmos que manipulam o conjunto de variáveis dentro e fora da base e requer o armazenamento de apenas uma matriz para o processo. Escrito via *tableau* após a multiplicação pela inversa da matriz B , o sistema anterior poderia ser colocado da forma:

	x_B	x_D	\bar{b}
x_B	I	$B^{-1}D$	$B^{-1}b$

onde tem-se na parte externa da caixa a legenda referente aos elementos da matriz. x_B como comentado, são as variáveis da base e estão associadas às colunas da matriz identidade, bem como as linhas do *tableau* que indicam as variáveis ativas no sistema. As variáveis fora da base x_D , estão associadas às colunas dos elementos da matriz $B^{-1}D$. Por fim, a coluna \bar{b} , denominada coluna à direita, contém o valor do vetor b alterado pelas operações realizadas na construção da base (B^{-1}).

Estipulando sobre o sistema a condição de não-negatividade das variáveis da base ($x_B \geq 0$) e compondo, por definição, para cada par (w_j, z_j) a inclusão de uma destas variáveis na base com a outra fora (condição de complementaridade), o LCP pode ser reformulado como a busca de uma base adequada para o sistema

subdeterminado descrito na Equação 2.3, de onde tem-se o algoritmo de Lemke.

$$\begin{aligned} \begin{bmatrix} I & -M \end{bmatrix} \begin{bmatrix} w \\ z \end{bmatrix} &= q \\ w \geq 0, z \geq 0 \\ w^T z &= 0 \end{aligned} \tag{2.3}$$

Em sua fase inicial, o algoritmo de Lemke verifica a existência de uma solução trivial $(w, z) = (q, 0)$, com $q \geq 0$. Caso a solução seja viável, o processo é finalizado tomando esta como solução do LCP. Caso a solução trivial não seja válida, busca-se viabilizar o sistema adicionando uma nova coluna $-e_p$ (elementos da coluna de valor -1), associada à variável artificial z_0 . O *tableau* é modificado, adquirindo a forma apresentada na Equação 2.4.

$$\begin{array}{c|cccc} & w & z & z_0 & \bar{q} \\ \hline w & I & -M & -e_p & q \end{array} \tag{2.4}$$

Com o acréscimo da variável z_0 e sua inserção na base através de operações de pivotamento, o sistema atende às restrições de não-negatividade e complementaridade mas torna-se dependente de uma variável artificial, sendo uma solução complementar quase-viável.

Para tornar esta solução complementar viável, busca-se retirar a variável z_0 da base através de trocas sucessivas de variáveis na mesma. Uma descrição do processo gerado pela implementação básica do Algoritmo de Lemke, como visto em Bazaraa et al. (2005), segue:

Passo 1: Se $q \geq 0$ pare; $(w, z) = (q, 0)$ é uma solução para o problema de complementaridade. Caso contrário, selecionar $q_s = \min\{q_i : i = 1, \dots, p\}$ e alterar o *tableau* pivotando a linha s e a coluna z_0 . Atribuir a variável auxiliar $y_s = z_s$. Executar Passo 2.

Passo 2: Seja d_s a coluna associada à variável y_s . Se $d_s \leq 0$, executar Passo 5. Caso contrário, determinar r via o teste de razão mínima, onde \bar{q} é a coluna à

direita atualizada pelas operações no *tableau*:

$$\frac{\bar{q}_r}{d_{rs}} = \min \left\{ \frac{\bar{q}_i}{d_{is}} : d_{is} > 0, i = 1, \dots, p \right\}$$

se a variável básica associada à linha r é z_0 , executar Passo 4. Caso contrário, executar Passo 3.

Passo 3: A variável básica associada à linha r ou é componente de w ou de z . A variável y_s entra na base e o *tableau* é pivotado entre a linha r e a coluna y_s . Se a variável que sai da base é componente de w , a variável complementar em z é associada a y_s . Se a variável que sai da base é componente de z , é feita a operação análoga em y_s . Executar Passo 2.

Passo 4: Se quando y_s entra na base, z_0 sai da base, então pivotar a coluna y_s com a linha r referente a z_0 , produzindo uma solução complementar viável. Pare.

Passo 5: Parada com terminação em raio. (w, z, z_0) é uma solução complementar quase-viável.

As operações utilizadas permitem que o sistema mantenha uma solução complementar quase-viável durante todo o processo: com a escolha via teste da razão mínima garante-se que, para cada mudança realizada na base, as variáveis da base se mantenham não negativas; da mesma forma, a escolha de remover da base a variável complementar à que foi inserida mantém a condição de complementaridade enquanto se busca retirar z_0 .

Ainda, o critério de parada em raio ocorre quando, para escolha de uma variável a entrar na base, a coluna que a referencia possui todos os elementos não-positivos. Esta condição leva a que, para qualquer escolha de variável a ser retirada da base, não haja limitação através do critério de razão mínima sendo a solução resultante traçada sobre uma meia-linha extrema (*extreme half-line*) ou uma fronteira ilimitada.

Neste caso, pode-se dizer que, para a escolha de M e q , o Algoritmo de Lemke não foi capaz de encontrar uma solução complementar viável, seja por esta solução não existir no LCP em questão, seja pelo algoritmo não ser capaz de alcançá-la,

não sendo possível, *a priori*, afirmar qual a situação em questão.

2.2.2 Exemplos de execução para o algoritmo de Lemke

Visando facilitar o entendimento do algoritmo de Lemke, seguem alguns exemplos de aplicação como apresentados em Bazaraa et al. (2005). A execução do algoritmo é mostrada em formato *tableau*, demonstrando as sucessivas operações de pivotamento através das iterações do algoritmo. Tem-se ainda, representado no lado externo da matriz e alinhado às linhas, quais variáveis estão inseridas na base, enquanto que alinhada às colunas pode-se observar as variáveis correspondentes a estas colunas.

Em um primeiro exemplo, contemplando uma execução correta do algoritmo de Lemke buscou-se uma solução para o par (q, M) apresentado a seguir:

$$M = \begin{bmatrix} 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & -2 \\ 1 & -1 & 2 & -2 \\ 1 & 2 & -2 & 4 \end{bmatrix}, \quad q = \begin{bmatrix} 2 \\ 2 \\ -2 \\ -6 \end{bmatrix}$$

Inicialização: Começando o processo verifica-se no vetor q a existência de elementos negativos, o que inviabiliza uma solução trivial, levando então à montagem do sistema em formato *tableau*, como visto a seguir:

	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	\bar{q}
w_1	1	0	0	0	0	0	1	1	-1	2
w_2	0	1	0	0	0	0	-1	2	-1	2
w_3	0	0	1	0	-1	1	-2	2	-1	-2
w_4	0	0	0	1	-1	-2	2	-4	$\textcircled{-1}$	-6

Neste primeiro momento, busca-se na coluna referente ao vetor \bar{q} (vetor q modificado) o elemento de valor mais negativo, pivotando o componente do vetor z_0 correspondente a esta linha na matriz. Esta operação corresponde à inserção da variável z_0 na base, substituindo a variável w_4 .

Iteração 1: Como a variável w_4 foi removida da base viável, busca-se na coluna referente à variável z_4 (*tableau* abaixo) o elemento positivo de menor razão quando relacionado à coluna \bar{q} (teste de razão mínima), selecionando a linha correspondente à variável w_3 através deste critério (razão $\frac{4}{6}$).

	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	\bar{q}
w_1	1	0	0	-1	1	2	-1	5	0	8
w_2	0	1	0	-1	1	2	-3	6	0	8
w_3	0	0	1	-1	0	3	-4	6	0	4
z_0	0	0	0	-1	1	2	-2	4	1	6

Desta forma, a variável z_4 entra na base substituindo a variável w_3 através da operação de pivotamento do elemento correspondente (destacado na matriz).

Iteração 2: Seguindo o mesmo procedimento, a remoção da variável w_3 da base viável leva a uma busca na coluna referente à variável z_3 para sua inserção. Novamente é realizado o teste da razão mínima, selecionando a linha correspondente à variável w_1 dado ser o elemento positivo de menor valor neste critério (razão $\frac{14/3}{7/3}$).

	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	\bar{q}
w_1	1	0	-5/6	-1/6	1	-1/2	7/3	0	0	14/3
w_2	0	1	-1	0	1	-1	1	0	0	4
z_4	0	0	1/6	-1/6	0	1/2	-2/3	1	0	2/3
z_0	0	0	-2/3	-1/3	1	0	2/3	0	1	10/3

A variável z_3 substitui então a variável w_1 na base através da operação de pivotamento no elemento correspondente.

Iteração 3: No próximo passo, tendo sido retirada a variável w_1 da base viável, busca-se a inserção da variável z_1 . Testa-se todos os elementos positivos na coluna referente a variável z_1 , selecionando a linha correspondente a variável z_0 (razão $\frac{2}{5/7}$).

	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	\bar{q}
z_3	3/7	0	-5/14	-1/14	3/7	-3/14	1	0	0	2
w_2	-3/7	1	-9/14	1/14	4/7	-11/14	0	0	0	2
z_4	2/7	0	-1/14	-3/14	2/7	5/14	0	1	0	2
z_0	-2/7	0	-3/7	-2/7	<u>5/7</u>	1/7	0	0	1	2

É realizada a operação de pivotamento entre o elemento da coluna z_1 e linha z_0 (elemento destacado). Como é verificada a remoção da variável artificial z_0 da base, o algoritmo de Lemke se encerra, tendo encontrado uma solução complementar viável.

Após a remoção da variável z_0 da base, o algoritmo produz em seu estado final o *tableau* apresentado a seguir:

	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	\bar{q}
z_3	3/5	0	-1/10	1/10	0	-3/10	1	0	-3/5	4/5
w_2	-1/5	1	-3/10	3/10	0	-9/10	0	0	-4/5	2/5
z_4	2/5	0	1/10	-1/10	0	3/10	0	1	-2/5	6/5
z_1	-2/5	0	-3/5	-2/5	1	1/5	0	0	7/5	14/5

É possível observar que as variáveis presentes na base são (z_3, w_2, z_4, z_1) , com valores respectivos apresentados na coluna \bar{q} . Desta forma, uma solução para o LCP apresentado é $w = (0, 2/5, 0, 0)^T$, $z = (14/5, 0, 4/5, 6/5)^T$.

No segundo exemplo é mostrado um caso de insucesso do algoritmo de Lemke, com ocorrência de uma parada em raio. Neste exemplo, busca-se uma solução do LCP produzido pelo par (q, M) apresentado a seguir:

$$M = \begin{bmatrix} 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 2 \\ -1 & 1 & 2 & -2 \\ 1 & -2 & -2 & 2 \end{bmatrix}, \quad q = \begin{bmatrix} 1 \\ 4 \\ -2 \\ -4 \end{bmatrix}$$

Inicialização: Como primeiro passo, o vetor q é analisado e, como há elementos negativos, não existe solução trivial, prosseguindo o algoritmo através da montagem do sistema em formato *tableau*, como visto a seguir:

	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	\bar{q}
w_1	1	0	0	0	0	0	-1	1	-1	1
w_2	0	1	0	0	0	0	1	-2	-1	4
w_3	0	0	1	0	1	-1	-2	2	-1	-2
w_4	0	0	0	1	-1	2	2	-2	$\textcircled{-1}$	-4

Busca-se então na coluna referente ao vetor \bar{q} (vetor q modificado) o elemento de valor mais negativo, pivotando o componente do vetor z_0 correspondente a esta linha na matriz (elemento destacado), o que corresponde à inserção da variável z_0 na base, substituindo, neste caso, a variável w_4 .

Iteração 1: Com o elemento w_4 removido da base, busca-se inserir o elemento z_4 de forma a preservar a condição de complementaridade (*tableau* abaixo). Através do teste da razão mínima entre os elementos das colunas \bar{q} e z_4 verifica-se que a razão positiva de menor valor corresponde à linha da variável w_3 (razão $\frac{2}{4}$).

	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	\bar{q}
w_1	1	0	0	-1	1	-2	-3	3	0	5
w_2	0	1	0	-1	1	-2	-1	0	0	8
w_3	0	0	1	-1	2	-3	-4	$\textcircled{4}$	0	2
z_0	0	0	0	-1	1	-2	-2	2	1	4

Dá-se então o pivotamento no elemento da coluna z_4 e linha w_3 (destacado na matriz), resultando na troca destas variáveis na base.

Iteração 2: Tendo sido o elemento w_3 removido da base, busca-se novamente, via teste da razão mínima, o elemento positivo da coluna z_3 de menor valor de razão. Verifica-se neste caso que todos os elementos na coluna referente a variável z_3 são não-positivos, inviabilizando a escolha de algum elemento para prosseguir a execução do algoritmo.

	w_1	w_2	w_3	w_4	z_1	z_2	z_3	z_4	z_0	\bar{q}
w_1	1	0	$-3/4$	$-1/4$	$-1/2$	$1/4$	0	0	0	$7/2$
w_2	0	1	0	-1	1	-2	-1	0	0	8
z_4	0	0	$1/4$	$-1/4$	$1/2$	$-3/4$	-1	1	0	$1/2$
z_0	0	0	$-1/2$	$-1/2$	0	$-1/2$	0	0	1	3

Neste caso, foi atingido o critério de parada em raio, não sendo possível para o algoritmo de Lemke apresentar solução para o LCP em questão.

2.2.3 Convergência do Algoritmo de Lemke

Apesar da indefinição relacionada ao critério de parada em raio, existem certas propriedades que podem ser exigidas sobre a matriz M garantindo a convergência durante uma execução. Seguem as Definições 1 e 2 e, em seguida, o Teorema 4, relativos a esta questão. O Resultado 1 traz ainda uma forma mais útil de análise sobre a matriz M .

Definição 1 Uma matriz quadrada M de ordem p é dita co-positiva se $y^T M y \geq 0$ para todo $y \geq 0$, $y \in \mathbf{R}^p$.

Definição 2 Uma matriz quadrada M de ordem p é dita co-positiva *plus* se ela é co-positiva e, para todo $y \geq 0$, $y \in \mathbf{R}^p$, quando se tem $y^T M y = 0$ segue-se que $(M + M^T)y = 0$.

Teorema 4 Se uma matriz M é co-positiva *plus* de ordem p , então, caso o LCP definido por M tenha solução, o Algoritmo de Lemke converge para esta. Caso o Algoritmo de Lemke termine com uma parada em raio, o LCP é inviável.

Resultado 1 Uma matriz quadrada M de ordem p positiva semi-definida é co-positiva *plus* e, caso haja solução viável para o LCP associado, o Algoritmo de Lemke irá convergir para tal solução, realizando uma parada em raio em caso contrário.

A prova deste teorema pode ser vista em (Murty, 1997). Como pode ser notado, o Teorema 4 não apresenta impedimento para execução do Algoritmo de Lemke em matrizes que não sejam co-positiva *plus*; apenas não possibilita que se tire alguma conclusão quanto à ocorrência de uma parada em raio, podendo, ainda assim, haver solução para o LCP. Em um contexto prático, o Algoritmo de Lemke pode ser utilizado mesmo que não hajam garantias plenas acerca de sua convergência (especialmente em se tratando de problemas onde o par (q, M) se altera a todo momento), dependendo esta escolha das necessidades de cada aplicação.

De fato, o único método conhecido capaz de garantir a existência ou não de solução para qualquer par (q, M) é a Enumeração dos Cones Complementares, que, como comentado anteriormente, apresenta complexidade exponencial (Murty, 1997).

2.3 O Problema de Programação em Dois Níveis

O problema de programação em dois níveis (abreviado aqui como BLPP) é um modelo de programação matemática desenvolvido para representar situações de competição com níveis hierárquicos de tomada de decisão. Pode ser definido como uma disputa entre dois agentes, o líder e o seguidor, com o líder tomando decisões em primeiro plano e o seguidor, por sua vez, definindo as escolhas que competem a ele a partir de decisões tomadas pelo líder.

O BLPP tem o que talvez seja sua primeira aparição documentada em um trabalho escrito pelo economista alemão Heinrich Freiherr von Stackelberg, onde é apresentado o problema econômico de Stackelberg para descrever situações de oligopólio de mercado (Stackelberg, 1952). Na definição, existiriam empresas con-

correntes em determinado mercado, sendo uma das empresas dominante neste mercado, denominada empresa líder, e tendo esta a capacidade de tomar decisões de maneira independente traçando, a partir de suas escolhas, um cenário para as empresas seguidoras. Essas, por sua vez, tomam decisões tendo em conta o cenário desenhado pela empresa líder com o objetivo de atender às suas próprias necessidades, podendo estas necessidades influenciar os desígnios do líder de maneira indireta.

No modelo de Stackelberg, o líder tem total conhecimento do comportamento do seguidor, o que o permite saber previamente as reações a partir de qualquer decisão tomada por si e possibilita o uso desta informação para atingir seus objetivos, já tendo em conta a situação final que encontrará. Da mesma forma, o seguidor tem total conhecimento da decisão tomada pelo líder e pode, baseado nesta informação, tomar a melhor decisão que atenda ao seus próprios interesses. Dadas essas características, o problema econômico de Stackelberg é denominado um jogo de informação completa.

Ainda que frequentemente correlacionados e sendo válidas as análises anteriores para ambos, existem algumas diferenças entre a formulação do problema econômico de Stackelberg e o BLPP, vinculadas especialmente ao escopo a que estão inseridos. Por exemplo, o problema econômico de Stackelberg, em princípio, não limita o número de passos possíveis entre líder e seguidor, até que estes atinjam seu ponto de equilíbrio, enquanto o BLPP permite apenas um passo podendo ser definido como uma versão estática do problema econômico de Stackelberg (Bard, 1998). Outro ponto é que, mesmo tendo o conceito de competição, o problema econômico de Stackelberg não exige que haja, necessariamente, um problema de minimização entre suas restrições, podendo este comportamento ser representado por outros tipos de problemas de programação matemática com restrições de equilíbrio (MPECs) (Colson et al., 2007), não sendo necessariamente representado por um BLPP.

Mesmo tendo aparecido inicialmente em 1934 no contexto do problema econô-

mico de Stackelberg, o BLPP só seria formalmente apresentado à comunidade científica em Bracken e McGill (1973, 1974), onde foi definido como um problema de otimização que contém, entre suas restrições, outro problema de otimização a ser resolvido. Esta formulação trouxe uma definição matemática mais rigorosa do problema, proporcionando o desenvolvimento deste campo e o separando em certa medida da esfera econômica.

O caso predominante, em um momento inicial, envolveu apenas funções lineares, sendo considerado mais simples por possuir alguns atributos úteis como a localização da solução ótima em um dos vértices da região viável. Desta forma, duas abordagens de destaque se basearam na enumeração de pontos extremos (Candler e Townsley, 1982; Bialas e Karwan, 1984) e na reformulação do problema via condições KKT, com posterior resolução da parte linear de forma exata e tratamento das condições de complementaridade via técnica *branch-and-bound* (Bard e Falk, 1982).

Algumas outras técnicas utilizadas para resolver o caso linear a serem comentadas são as baseadas em funções de penalização (Anandalingam e White, 1990; White e Anandalingam, 1993) e pivotamento complementar (Júdice e Faustino, 1988, 1992). Apesar de apresentar a menor complexidade entre as versões de BLPP, Hansen et al. (1992) demonstrou que, mesmo no caso linear, não é possível elaborar um algoritmo determinístico em tempo polinomial capaz de fornecer a solução ótima, sendo o problema comprovadamente NP-Difícil.

Com o desenvolvimento de abordagens de resolução para o BLPP, funções com outras características tiveram lugar. Algoritmos para resolução de funções não-lineares muitas vezes se limitam ao caso convexo devido às boas características que as funções apresentam (de forma análoga às funções lineares) e, mesmo aqueles que lidam com não-convexidade, se restringem, em geral, à busca de pontos estacionários e mínimos locais. Entre estas técnicas é relevante citar aquelas baseadas em *branch-and-bound* (Bard, 1988), métodos de descida (Savard e Gauvin, 1994), funções de penalização (Aiyoshi e Shimizu, 1984) e regiões de confiança

(Colson et al., 2005).

A resolução de problemas de programação inteira em dois níveis também está presente na literatura (Wen e Yang, 1990; Bard e Moore, 1992), porém apresentando maior desafio pelas complicações inerentes à programação inteira.

Recentemente, tendo em vista a dificuldade na resolução de instâncias do BLPP, especialmente aquelas envolvendo funções com maior grau de complexidade analítica (como em funções não-lineares, ou mesmo não-diferenciáveis), o uso de metaheurísticas se apresentou como alternativa promissora. Mais comentários sobre este tipo de abordagem serão apresentadas na Subseção 2.3.4.

2.3.1 Definição Matemática

Como comentado anteriormente, problemas de otimização em dois níveis se caracterizam por, além das características usuais de problemas de otimização, possuírem outra função a ser otimizada no nível de restrição. Apesar desta novidade aparentemente não representar uma alteração radical, a adição da nova função objetivo faz com que o funcional a ser otimizado apresente comportamento muito mais imprevisível (mesmo para funções suaves), o que limita quais propriedades possam ser esperadas e resulta em um correspondente aumento de complexidade.

De forma geral, pode-se descrever um problema de otimização em dois níveis como na Equação 2.5.

$$\begin{aligned}
 & \min_x \quad F(x, y) \\
 & \text{s.t.} \quad G(x, y) \leq 0 \\
 & \quad \min_y \quad f(x, y) \\
 & \quad \text{s.t.} \quad g(x, y) \leq 0
 \end{aligned} \tag{2.5}$$

com $x \in \mathbf{R}^{n_1}$, $y \in \mathbf{R}^{n_2}$ sendo as variáveis de líder e seguidor, respectivamente. Ainda, $F(x, y) : \mathbf{R}^n \rightarrow \mathbf{R}$ e $f(x, y) : \mathbf{R}^n \rightarrow \mathbf{R}$, $n = n_1 + n_2$, são as funções objetivo de líder e seguidor e $G(x, y) : \mathbf{R}^n \rightarrow \mathbf{R}^{m_1}$, $g(x, y) : \mathbf{R}^n \rightarrow \mathbf{R}^{m_2}$, as restrições de líder e seguidor, sendo m_1 e m_2 relacionados ao número de restrições dos respectivos

níveis e $m = m_1 + m_2$ ao número de restrições do problema.

Este trabalho busca tratar uma classe específica dessa formulação, quando a função $f(x, y)$ é quadrática e as restrições são lineares em relação as variáveis do seguidor, y . Desta forma, o problema apresentado anteriormente (Equação 2.5) pode ser redefinido. A Equação 2.6 apresenta o problema tratado neste trabalho.

$$\begin{aligned}
& \min_x \quad F(x, y) \\
& \text{s.t.} \quad G(x, y) \leq 0 \\
& \min_y \quad f(x, y) = \frac{1}{2}y^T Q(x)y + c(x)^T y + d(x) \\
& \text{s.t.} \quad A(x)y + b(x) \leq 0
\end{aligned} \tag{2.6}$$

Nesta formulação, as mudanças aparecem apenas no nível do seguidor. $Q(x) \in \mathbf{R}^{n_2 \times n_2}$ é uma matriz quadrada e simétrica, relacionada ao termo quadrático em y , dependente do valor de x . Da mesma forma $c(x), d(x) \in \mathbf{R}^{n_2}$ são responsáveis pela parte linear e constante da equação do seguidor, ambas dependentes de x . Para as restrições a matriz $A(x) \in \mathbf{R}^{n_2 \times m_2}$ e o vetor $b(x) \in \mathbf{R}^{m_2}$ estão relacionados ao multiplicador de y e termo constante respectivamente, ainda com dependência de x . Q é considerada simétrica sem risco de perda de generalidade, dado que, para Q não-simétrica tem-se $Q' = \frac{1}{2}(Q + Q^T)$, sendo Q' simétrica e a função $f(x, y)$ associada a $Q'(x)$ idêntica ao problema original (Murty, 1997).

Repare que a dependência em relação à variável x , presente em todos os componentes do nível do seguidor, não possui restrição quanto à sua forma, podendo aparecer de outras maneiras que não o caso linear ou quadrático (mesmo funções não-diferenciáveis são válidas nesta formulação). Para o nível do líder não há nenhuma mudança, mantendo-se a formulação geral com quaisquer $F(x, y)$ e $G(x, y)$ sendo válidos.

Inspirado em Bard (1998), segue uma maneira útil de se determinar as relações de domínio do problema em dois níveis com função quadrática e restrições lineares no nível do seguidor. $X \subset \mathbf{R}^{n_1}$ e $Y \subset \mathbf{R}^{n_2}$ são os subespaços definidos para as variáveis de líder e seguidor respectivamente.

- O espaço viável a partir das restrições de líder e seguidor.

$$S = \{(x, y) : x \in X, y \in Y, G(x, y) \leq 0, A(x)y + b(x) \leq 0\}$$

- O espaço viável para o seguidor.

$$S(x) = \{y : y \in Y, A(x)y + b(x) \leq 0\}$$

- A projeção da região viável S no espaço de decisão do líder.

$$S(X) = \{(x, y) : x \in X, \exists y \in Y, G(x, y) \leq 0, A(x)y + b(x) \leq 0\}$$

- A reação racional do seguidor para $x \in S(X)$.

$$P(x) = \{y : y \in \operatorname{argmin}\{f(x, \hat{y}) = \frac{1}{2}\hat{y}^T Q(x)\hat{y} + c(x)^T \hat{y} + d(x), \hat{y} \in S(x)\}\}$$

- A reação induzida ou conjunto viável do problema.

$$IR = \{(x, y) : (x, y) \in S, y \in P(x)\}$$

Utilizando esta notação é possível perceber que para uma mesma solução do líder, o conjunto de reação racional $P(x)$ pode não ser unitário, levando a considerações sobre como o seguidor pode reagir a múltiplas escolhas. Outro problema ocorre quando $P(x) = \emptyset$, não havendo neste caso maneira de se determinar uma solução para o líder. Para evitar estes problemas é comum na literatura a consideração de que $P(x)$ é unitário em todo valor de x , mantendo assim o problema bem-posto. Estes pontos são melhor discutidos na Subseção 2.3.2.

2.3.2 Análise de otimalidade no BLPP

Existência e otimalidade de soluções são conceitos de difícil definição ao serem aplicados no BLPP. Mesmo nos casos mais simples, de funções e restrições contínuas em conjuntos compactos, não é possível garantir que todas as configurações possíveis para a variável do líder possuam resposta válida no seguidor ou mesmo sua unicidade (Bard, 1998).

No caso específico deste trabalho, é relevante analisar a situação quando há uma função quadrática no nível do seguidor e restrições lineares. Inicialmente, replicando a Equação 2.6 apenas para o seguidor, tem-se a formulação básica de um problema quadrático com restrições lineares (Equação 2.7).

$$\begin{aligned} \min_y \quad & f(x, y) = \frac{1}{2}y^T Q(x)y + c(x)^T y + d(x) \\ \text{s.t.} \quad & A(x)y + b(x) \leq 0 \end{aligned} \tag{2.7}$$

A função $f(x, y)$ é convexa se e somente se $Q(x)$ é positiva semi-definida, podendo ser estritamente convexa caso esta matriz seja positiva definida. Estas são condições de convexidade sobre todo o espaço \mathbf{R}^{n_2} , podendo ser verificadas situações em que há convexidade para $f(x, y)$ apenas no subespaço de \mathbf{R}^{n_2} definido pelas restrições aplicadas ao problema (Murty, 1997).

A convexidade é condição desejável para a resolução, por possibilitar a utilização de diversas técnicas que tem como base esta condição, além de garantir a existência de um único mínimo, sendo este mínimo global. De fato, quando o problema de programação quadrática trata uma função não-convexa, este passa a ser classificado como NP-Difícil (Murty e Kabadi, 1987), o que por si só apresenta desafios próprios.

Quando $Q(x)$ é positiva definida e o subespaço $S(x)$ não-vazio para todo valor possível de x , existirá uma aplicação unívoca em $P(x)$ levando sempre a um único valor de y no problema de minimização (Vicente, 1992). Estas condições podem não ser atendidas, o que não impede a existência de solução para o problema geral mas gera a necessidade de estratégias para se tratar inconsistências, tanto relativas

a múltiplas soluções no seguidor, quanto à inexistência destas.

No caso de existência de múltiplas soluções para o seguidor, uma análise recorrente na literatura é através das abordagens otimista versus pessimista. Na abordagem otimista, espera-se uma situação de colaboração ou controle entre líder e seguidor, com o seguidor tomando, em caso de múltiplas opções ótimas, sempre aquela que leve a um maior retorno para o líder. Já na abordagem pessimista, espera-se uma situação de competição acirrada ou de minimização de riscos para o líder, o que faz com que este sempre tome por escolha do seguidor a solução ótima que leve à maior perda para o líder.

O caso de inexistência de soluções para o seguidor é menos explorado na literatura, sendo prática comum a simples penalização das configurações do líder com esta condição, considerando-as inviáveis e com função objetivo arbitrariamente grande.

Neste contexto, o trabalho aqui proposto não tem como objetivo investigar técnicas para tratar casos com múltiplas soluções ou inexistência das mesmas (dadas as próprias limitações na resolução via método de Lemke para o seguidor), mas evita a utilização de pressupostos que garantam a unicidade de $P(x)$ buscando não limitar seus casos de aplicação. Maiores detalhes acerca de como estes dois pontos são tratados podem ser vistos na descrição da abordagem (Capítulo 4).

2.3.3 O BLPP como um problema de complementaridade linear

Como apresentado anteriormente, uma maneira comum de se reescrever o BLPP é através das condições KKT. Estas permitem, com a adição de novas variáveis e restrições, a elaboração do BLPP como um modelo de otimização de um

único nível. A formulação segue na Equação 2.8.

$$\begin{aligned}
& \min_{x,y,\lambda} && F(x,y) \\
& \text{s.t.} && G(x,y) \leq 0 \\
& && \nabla_y f(x,y) + \sum_i \lambda_i \nabla_y g_i(x,y) = 0 \\
& && \lambda_i g_i(x,y) = 0 \\
& && g(x,y) \leq 0 \\
& && \lambda \geq 0
\end{aligned} \tag{2.8}$$

Considerando as restrições impostas à função e as restrições do seguidor comentadas anteriormente, pode-se reorganizar o problema de forma a simplificar a notação.

$$\begin{aligned}
& \min_{x,y,\lambda} && F(x,y) \\
& \text{s.t.} && G(x,y) \leq 0 \\
& && Q(x)y + A(x)^T \lambda + c(x) = 0 \\
& && A(x)y + b(x) + u = 0 \\
& && \lambda_i u_i = 0 \\
& && \lambda \geq 0, u \geq 0
\end{aligned} \tag{2.9}$$

onde u é uma variável de folga utilizada para transformar as restrições de desigualdade em restrições de igualdade.

Este problema, sob determinadas condições, é equivalente ao problema de minimização presente na Equação 2.6. A seguir, os Teoremas 5 e 6, apresentados em Murty (1997), fornecem condições de equivalência para o problema quadrático no nível do seguidor (Equação 2.7) e a formulação utilizando o KKT em Equação 2.9.

Teorema 5 Se \bar{y} é solução ótima para o problema Equação 2.7, então \bar{y} é um ponto que atende às condições KKT, seja $Q(x)$ positiva semi-definida ou não.

Teorema 6 Se $Q(x)$ é positiva semi-definida e \bar{y} é um ponto que atende às condições KKT, então \bar{y} é solução ótima do problema de minimização Equação 2.7.

A partir destes teoremas tem-se que, caso a matriz $Q(x)$ seja positiva semi-definida, o problema presente na Equação 2.6 é equivalente à sua reescrita no nível do seguidor através das condições KKT presentes na Equação 2.9. Caso contrário, o ponto ótimo \bar{y} obedece às condições KKT, mas não é possível garantir que seja o único a obedecê-las.

Esta análise está vinculada à convexidade da função $f(x, y)$, sendo relevante notar, como comentado na Subseção 2.3.2, que esta função pode ser convexa em outras situações dado o subespaço definido pelas restrições, podendo levar inclusive a situações de equivalência com as condições KKT.

Inspirado no trabalho de Li e Wang (2006), é possível transformar a notação da Equação 2.9 em um problema de otimização com um problema de complementaridade linear entre suas restrições. Para isso, deve-se impor uma condição de não-negatividade ao vetor y ($y \geq 0$). A reescrita deste problema pode ser vista na Equação 2.10.

$$\begin{aligned}
& \min_{x,y,\lambda} && F(x, y) \\
& \text{s.t.} && G(x, y) \leq 0 \\
& && \lambda'' - (Q(x)y + A(x)^T \lambda') = c(x) \\
& && u - (-A(x)y + 0\lambda') = -b(x) \\
& && \lambda'_i u_i = 0 \\
& && \lambda''_j y_j = 0 \\
& && \lambda' \geq 0, \lambda'' \geq 0, u \geq 0 \\
& && y \geq 0
\end{aligned} \tag{2.10}$$

onde o vetor de multiplicadores (λ) foi dividido em duas partes: uma referente às restrições de desigualdade no nível do seguidor (λ') e a outra referente às condições de não negatividade impostas a y (λ''). É possível associar então os elementos aos componentes do LCP:

$$w = \begin{pmatrix} \lambda'' \\ u \end{pmatrix}, z = \begin{pmatrix} y \\ \lambda' \end{pmatrix}$$

$$M = \begin{pmatrix} Q & A^T \\ -A & 0 \end{pmatrix}, q = \begin{pmatrix} c \\ -b \end{pmatrix}$$

possibilitando a resolução do nível do seguidor através da resolução do LCP definido por estas relações. Mais precisamente, o problema de encontrar o ponto y e multiplicadores λ' , λ'' representado pelas condições KKT é substituído pelo LCP apresentado. Portanto se \bar{y} é solução ótima do problema quadrático do seguidor, existe uma escolha de valores para λ' , λ'' e u que combinados a $y = \bar{y}$ formam um par (w, z) solução para o LCP. Mais detalhes sobre a escrita de um problema quadrático como um LCP podem ser vistos em Murty (1997). Note ainda que a reescrita não altera quaisquer questões no que tange à possível existência de múltiplas soluções para as condições KKT, sendo possível, para funções não-convexas, encontrar pontos via resolução do LCP que obedeçam às condições KKT mas não sejam ótimos.

Em princípio, a notação anterior exige que o vetor y apenas assuma valores positivos, o que poderia restringir o escopo da abordagem. Mas esta dificuldade é facilmente contornável através da reescrita das variáveis y que podem assumir valores negativos pela expressão $y = y' - y''$, onde $y', y'' \geq 0$.

Da mesma forma, problemas com restrições de igualdade no nível do seguidor poderiam ser facilmente resolvidos pela substituição de cada restrição de igualdade por duas restrições de desigualdade, uma de não-negatividade e outra de não-positividade, desde que as restrições de igualdade sejam lineares. Estas duas formulações, apesar de expandirem o escopo da técnica, tem como desvantagem o aumento da dimensão do problema, resultando em mais processamento para a resolução do LCP.

Por fim, considerando a utilização do Algoritmo de Lemke para garantia da convergência deve-se analisar a matriz M obtida pela composição do problema. Como comentado anteriormente, uma condição suficiente está relacionada a esta

matriz ser positiva semi-definida. Analisando essa possibilidade tem-se:

$$\begin{aligned}
\begin{bmatrix} u & v \end{bmatrix}^T \begin{bmatrix} Q & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} &\geq 0 \\
\begin{bmatrix} u^T Q - v^T A & u^T A^T \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} &\geq 0 \\
u^T Q u - v^T A u + u^T A^T v &\geq 0 \\
u^T Q u - v^T A u + (A u)^T v &\geq 0 \\
u^T Q u - v^T A u + (v^T A u)^T &\geq 0 \\
u^T Q u &\geq 0
\end{aligned} \tag{2.11}$$

Sendo portanto suficiente, para garantia de convergência do Algoritmo de Lemke, a exigência de que a matriz Q seja positiva semi-definida. É possível ainda que, para aquelas funções convexas restritas ao domínio definido pelas restrições, o Algoritmo de Lemke seja capaz de convergir, encontrando uma solução ótima, mas isto não pode ser garantido, podendo uma parada em raio ocorrer mesmo havendo solução.

Este mesmo método pode ser utilizado para resolução de funções lineares no nível do seguidor, sendo um caso particular com matriz $Q = 0$.

Ainda, uma dificuldade deste método está na definição de qual valor do seguidor será encontrado, quando há múltiplos pontos que são ótimos globais, não sendo possível definir uma abordagem pessimista ou otimista pelo simples fato da indefinição de qual ponto será encontrado pelo Algoritmo de Lemke.

2.3.4 Aplicação de Metaheurísticas ao BLPP

Nos últimos anos a aplicação de metaheurísticas a problemas de programação em dois níveis tem se tornado uma ativa área de pesquisa (Talbi, 2013). O interesse vem de encontro às dificuldades presentes no desenvolvimento de estratégias de resolução determinísticas para o BLPP, especialmente em casos não-lineares e com variáveis inteiras.

Apesar de se apresentar como uma abordagem promissora, o uso de metaheurísticas vinculadas ao BLPP apresenta alguns desafios em sua implementação. O principal destes está na dificuldade em avaliar a função objetivo para o líder, dado que, *a priori*, é necessária a solução exata do problema de minimização no nível do seguidor, o que pode não ser garantido com uma metaheurística. Mesmo sendo uma dificuldade a ser enfrentada, Angelo e Barbosa (2015) demonstrou que, tendo-se os cuidados necessários com a convergência do seguidor, a utilização de soluções aproximadas é viável de um ponto de vista prático.

Outro fator relevante é a não-convexidade do BLPP, que cria dificuldades na exploração do espaço de busca e convergência, sendo necessário determinado esforço para garantir uma adequada análise da região.

Segundo Talbi (2013) estratégias metaheurísticas para o BLPP podem ser classificadas em quatro tipos:

- Abordagem aninhada: em cada solução avaliada para o líder, é resolvido um problema de otimização aninhado para o seguidor de forma a fornecer a resposta mais correta deste.
- Abordagem de transformação em um único nível: o BLPP é alterado de forma a representar a competição como uma única função objetivo, estabelecendo o problema através da formulação tradicional em otimização.
- Abordagem multiobjetivo: o BLPP é alterado de forma a representar a competição através de uma formulação multiobjetivo.
- Abordagem co-evolutiva: são desenvolvidas múltiplas heurísticas de forma a evoluir os níveis do problema em paralelo, ocorrendo compartilhamento de informação entre estas heurísticas.

A abordagem aninhada é, provavelmente, a maneira imediata de se imaginar a resolução de um problema em dois níveis. Entre os trabalhos que se valem deste método há aqueles que, considerando a existência de técnicas eficazes de resolução

exata para algum dos níveis, combinam esta técnica com uma metaheurística o que pode levar a bons resultados (Li e Wang, 2006; Li e Zhang, 2014). Outros trabalhos procuram resolver ambos os níveis através de metaheurísticas, buscando critérios de parada que permitam boa convergência no seguidor (Angelo et al., 2013).

Abordagens que transformam o BLPP em um problema equivalente em um único nível se utilizam principalmente das condições KKT. Seu atrativo está em permitir, com esta mudança, a utilização dos diversos métodos desenvolvidos para a formulação tradicional de problemas de otimização (Li e Wang, 2010).

Abordagens multiobjetivo podem ser vistas em alguns trabalhos (Wang et al., 2005), sendo outra forma de se valer de técnicas de otimização de áreas distintas para resolução do BLPP.

Já as abordagens co-evolutivas são técnicas consideradas de propósito mais geral, sendo aparentemente promissoras e tendo a atenção de alguns autores que se dedicaram ao desenvolvimento de trabalhos neste contexto (Oduguwa e Roy, 2002; Legillon et al., 2012).

Por fim, é relevante citar a possibilidade de utilização de *matheuristics* na resolução do BLPP. *Matheuristics* se utilizam da capacidade de generalização das metaheurísticas com o bom desempenho de métodos exatos, combinando estas duas características no desenvolvimento de um método único. Esta combinação se apresenta como uma boa alternativa quando se considera o BLPP tendo em vista seu caráter aninhado, o que permite, por exemplo, a resolução de cada um dos níveis com uma metodologia diferente. Neste contexto, podem ser citadas algumas iniciativas como: novamente o trabalho de Li e Wang (2006), que utiliza (assim como o trabalho aqui proposto) do algoritmo de Lemke para resolução do problema do seguidor, enquanto uma implementação de algoritmo genético é aplicada no problema do líder e, da mesma forma, o trabalho de Li e Zhang (2014) pode ser enquadrado nesta classe, se utilizando de uma implementação da metaheurística evolução diferencial para resolver o nível do líder, enquanto o nível do seguidor é delegado a um resolvidor matemático.

Capítulo 3

Evolução Diferencial

Pertencente ao escopo das abordagens evolutivas, a técnica de Evolução Diferencial (conhecida como DE) foi apresentada em Storn e Price (1995), sendo descrita como uma metaheurística simples e com bom desempenho em uma grande variedade de problemas.

De fato, a DE tem como seu principal atrativo ser um algoritmo de fácil implementação, especialmente quando comparado a outras estratégias evolutivas como o Algoritmo Genético, apresentando operações simples e executáveis em poucas linhas de código. Outro fator de interesse é o baixo número de parâmetros, o que permite um bom ajuste da técnica sem necessidade de demasiado esforço em testes de desempenho.

Mesmo com essa simplicidade, a DE vem se destacando e despertando o interesse da comunidade científica muito por sua grande competitividade em diversas aplicações, tendo destaque inclusive em competições entre metaheurísticas (Das e Suganthan, 2011).

O termo Evolução Diferencial é relativo à operação de mutação presente na técnica, onde são utilizadas diferenças entre vetores para definir direções de perturbação em um vetor base. Esta é a principal operação na geração de novas soluções (denominadas indivíduos neste contexto) em uma implementação de DE, sendo o grande ponto de diferenciação da técnica quando comparada a outras estratégias evolutivas.

Tratando-se de uma abordagem evolutiva, a DE se utiliza de conceitos presentes na teoria da evolução natural de Charles Darwin. Desta forma, é esperado que possua um conjunto de características baseadas no processo de seleção natural, sendo as principais: uma população de indivíduos; um processo de geração de novos indivíduos através de características já existentes na população; um processo de inserção de novas características na população; um processo de competição entre indivíduos para seleção parental e um processo de pressão evolutiva definindo maior probabilidade de manutenção para indivíduos mais aptos da população (Talbi, 2009). A definição de como estes pontos estão aplicados em uma implementação da DE serão melhor descritos nas subseções a seguir.

3.1 Fluxo Principal

Como comentado, a DE é um algoritmo de fácil implementação. Isto pode ser verificado no Algoritmo 1, onde é apresentado um esquema para implementação da técnica.

Algorithm 1 Estrutura da DE

```

1: function DE( $F$ ,  $CR$ , NP)
2:    $X \leftarrow$  InicializaPopulação(NP);
3:    $G \leftarrow 0$ ;
4:   while Critério de parada do
5:      $G \leftarrow G + 1$ ;
6:     for  $i \leftarrow 1 \dots NP$  do
7:       individuosMutação  $\leftarrow$  SelecionaIndivíduos( $X_G$ );
8:        $x_{MUT} \leftarrow$  Mutação(individuosMutação,  $F$ );
9:        $x_{REC} \leftarrow$  Recombinação( $x_{i,G}$ ,  $x_{MUT}$ ,  $CR$ );
10:      if  $f(x_{REC}) \leq f(x_{i,G})$  then
11:         $x_{i,G+1} \leftarrow x_{REC}$ 
12:      else
13:         $x_{i,G+1} \leftarrow x_{i,G}$ 
14:      end if
15:    end for
16:  end while
17:  return melhor elemento em  $X$ 
18: end function

```

Pode-se identificar via Algoritmo 1 a localização de operações comuns a algoritmos evolutivos na DE: seleção (Linha 7), mutação (Linha 8), recombinação

(Linha 9) e avaliação (Linhas 10 a 14). É a definição das operações de mutação e recombinação que, em geral, definirá a variante da DE utilizada, sendo a rotina de seleção dependente do operador de mutação utilizado.

A execução da DE começa com um processo de inicialização da população, que deve gerar diversidade suficiente para a convergência do algoritmo. Como próxima etapa, para cada indivíduo presente na população, um novo será gerado. Esta proporção é importante pois, no processo de avaliação para a próxima geração, a comparação ocorre dois a dois entre um elemento da geração anterior e um novo candidato a entrar na população. Para se criar um novo indivíduo, inicialmente são selecionados vetores da geração atual, de acordo com o processo de mutação, tendo então seus valores combinados de forma a gerar um vetor mutado (x_{MUT}). Este então sofrerá a operação de recombinação, tendo seus valores intercalados à um vetor de referência (x_i), produzindo o vetor re combinado (x_{REC}). Por fim, o vetor de referência utilizado na recombinação será comparado com o vetor re combinado através de uma função de avaliação, com o mais apto (de menor valor funcional no caso de minimização) sendo mantido na população.

Ainda, os parâmetros presentes no algoritmo são: F a ponderação do vetor diferença na operação de mutação, CR a probabilidade de utilização de novas características vindas da mutação no processo de recombinação e NP o número de indivíduos presentes na população. O vetor x , denominado vetor paramétrico, contém as variáveis de decisão que necessitam ser otimizadas.

Para facilitar a visualização de cada uma destas operações comentadas e a utilização dos parâmetros, uma implementação da versão básica da DE (DE/rand/1/bin) pode ser vista no Algoritmo 2.

Nesta implementação a população é inicializada aleatoriamente e, para cada indivíduo da nova população, são sorteados três indivíduos da população corrente (Linha 7) com estes fazendo parte da operação de mutação (Linha 8). Após geração do vetor mutado, este é re combinado com o vetor de referência na população corrente (Linhas 10-17), tendo por fim vetor de referência e vetor re combinado

Algorithm 2 DE Básica

```
1: function DE( $F, CR, NP$ )
2:    $X \leftarrow \text{InicializaPopulaçãoAleatoriamente}(NP)$ ;
3:    $G \leftarrow 0$ ;
4:   while Critério de parada do
5:      $G \leftarrow G + 1$ ;
6:     for  $i \leftarrow 1 \dots NP$  do
7:        $\{x_{r1,G}, x_{r2,G}, x_{r3,G}\} \leftarrow \text{SelecionaIndivíduos}(X_G)$ ; /*  $r_1 \neq r_2 \neq r_3$  */
8:        $x_{MUT} \leftarrow x_{r1,G} + F(x_{r2,G} - x_{r3,G})$ ;
9:        $jrand \leftarrow \text{rand}(1 \dots N)^1$ ; /*  $N$  dimensão do vetor paramétrico */
10:      for  $j \leftarrow 1 \dots N$  do
11:         $S \leftarrow \text{rand}(1 \dots 100)$ ;
12:        if  $S \leq CR \parallel jrand = j$  then
13:           $x_{REC}^j \leftarrow x_{MUT}^j$ 
14:        else
15:           $x_{REC}^j \leftarrow x_{i,G}^j$ 
16:        end if
17:      end for
18:      if  $f(x_{REC}) \leq f(x_{i,G})$  then
19:         $x_{i,G+1} \leftarrow x_{REC}$ 
20:      else
21:         $x_{i,G+1} \leftarrow x_{i,G}$ 
22:      end if
23:    end for
24:  end while
25:  return melhor elemento em  $X$ 
26: end function
```

comparados, com manutenção do de menor valor na função de avaliação (Linhas 18-22). O algoritmo termina através de algum critério de parada de interesse.

Outro ponto a ser comentado é relacionado à nomenclatura dada às variantes da DE. Tendo em vista que as operações principais ao se definir uma implementação desta metaheurística são mutação e recombinação, a nomenclatura tradicional identifica justamente a escolha destas operações na definição do algoritmo, seguindo a estrutura:

DE/vetor base/número de diferenças/recombinação

No primeiro termo define-se o método de seleção do vetor base onde ocorrerá a operação de mutação, podendo ser uma escolha aleatória (rand), um elemento

¹ $\text{rand}()$: função que retorna um número aleatório uniformemente distribuído num conjunto ou intervalo pré-fixados.

de referência da população atual (target), a melhor solução encontrada (best) ou o direcionamento do vetor base rumo a algum outro elemento (target-to-rand, rand-to-best). O segundo termo, apresentado após a barra, representa o número de perturbações realizadas pelas diferenças, sendo mais comum a escolha entre uma ou duas perturbações. O terceiro elemento comum na identificação de uma DE representa o método de recombinação, sendo as opções mais comuns a recombinação binomial e exponencial.

A seguir são descritas as fases presentes no fluxo de execução de uma implementação da DE.

3.2 Inicialização

A definição de como ocorrerá o processo de inicialização da população é dependente da aplicação a que a DE é voltada. A forma mais tradicional ocorre via inicialização aleatória dos vetores, gerando um conjunto suficientemente diverso para que se comece o processo de evolução.

Apesar de simples, esta nem sempre é uma escolha possível, quando se requer a satisfação de certas restrições pelos indivíduos iniciais, por exemplo, sendo necessário um procedimento mais apurado que ofereça essa garantia.

Da mesma forma, pode ser uma boa alternativa, para problemas com boas técnicas de construção de soluções, a utilização destas técnicas intercaladas com elementos aleatórios, de forma que a convergência possa ser acelerada sem grande custo adicional em recursos.

3.3 Mutação

A principal operação presente em uma implementação da técnica DE, a mutação ocorre tradicionalmente através da perturbação de um vetor base em que é adicionada a diferença entre outros vetores da população ponderada por um fator F . O denominado vetor base é aquele que receberá as perturbações, servindo como ponto de partida para a mutação.

Baseado no conceito da operação, diversas variantes são apresentadas na literatura, indo desde a variação básica, conhecida como rand/1, até outras que incluem informações das melhores soluções encontradas, como best-to-rand/1.

A escolha do processo de mutação talvez seja o que mais diferencia as implementações da DE, sendo a melhor opção dependente tanto da forma que o problema se apresenta quanto da própria implementação do algoritmo, considerando critério de parada, tamanho da população entre outros fatores.

Seguem alguns modelos de mutação utilizados na literatura:

- rand/1

$$x_f = x_{r1} + F(x_{r2} - x_{r3})$$

- target/1

$$x_f = x_i + F(x_{r1} - x_{r2})$$

- best/1

$$x_f = x_{best} + F(x_{r1} - x_{r2})$$

- rand-to-best/1

$$x_f = x_{r1} + F(x_{best} - x_{r1}) + F(x_{r2} - x_{r3})$$

- target-to-best/1

$$x_f = x_i + F(x_{best} - x_i) + F(x_{r1} - x_{r2})$$

- rand/2

$$x_f = x_{r1} + F(x_{r2} - x_{r3}) + F(x_{r4} - x_{r5})$$

- target/2

$$x_f = x_i + F(x_{r1} - x_{r2}) + F(x_{r3} - x_{r4})$$

- best/2

$$x_f = x_{best} + F(x_{r1} - x_{r2}) + F(x_{r3} - x_{r4})$$

onde os vetores x_{r1}, x_{r2}, \dots são indivíduos escolhidos aleatoriamente sem repetição, x_i é o vetor de referência utilizado para os processos de recombinação e avaliação e o vetor x_{best} é relativo ao melhor indivíduo presente na população.

Estes são apenas uma amostra dos possíveis operadores para mutação, sendo válida a utilização de quaisquer opções de estrutura semelhante, desde que envolvam operações de diferença entre vetores.

Uma dificuldade presente no processo de mutação é que a operação realizada pode levar elementos do vetor paramétrico a faixas de valores que estão fora das restrições do modelo. Desta forma é comum adicionar ao processo rotinas que lidem com este tipo de restrição, também denominada restrição de “caixa”.

A forma mais comum é a projeção dos elementos que ultrapassem a “caixa” viável (região formada pela combinação das restrições de “caixa”) no ponto extremo desta “caixa” através do seguinte esquema: no limite superior quando é ultrapassado o maior valor possível do elemento, o elemento recebe este maior valor possível, enquanto no limite inferior o procedimento é análogo trocando o termo maior por menor. A desvantagem deste método é uma possível perda de direções de diferença devido a alteração dos vetores, tendo diferentes violações levadas a um mesmo ponto.

Outra forma possível é a ponderação do vetor gerado pela diferença de forma que a adição deste ao vetor base se mantenha dentro da “caixa” viável. Esta forma contorna o problema gerado por diferentes direções levadas a um mesmo ponto, mas a depender do deslocamento, pode gerar alterações muito pequenas e, consequentemente, soluções muito próximas à solução base que não apresentam novidade ao processo evolutivo.

Uma ultima possibilidade é não limitar os vetores à “caixa” viável, mas tratar estas restrições de “caixa” através de procedimentos de penalização que impactem as violações diretamente no valor da função objetivo. Esta prática tem como

vantagem permitir uma maior liberdade de deslocamento para os vetores da população, mas faz com que estes vetores visitem regiões inviáveis desnecessariamente, desperdiçando, *a priori*, alguns recursos.

3.4 Recombinação

A operação de recombinação busca trazer maior diversidade através da combinação de características do antigo elemento da população e do novo gerado pelo processo de mutação.

Para isto, é feita a interseção entre a configuração destes dois indivíduos utilizando um processo aleatório de escolha. As maneiras mais comumente vistas são aquelas que se utilizam da distribuição aleatória binomial e da distribuição aleatória exponencial. Os Algoritmos 3 e 4 apresentam a implementação do processo de recombinação binomial e exponencial respectivamente.

Algorithm 3 Recombinação Binomial

```

1: function DE( $X_i$ ,  $X_{MUT}$ ,  $CR$ )
2:   jrand  $\leftarrow$  rand( $1 \dots N$ ); /* N dimensão do vetor paramétrico */
3:   for  $j \leftarrow 1 \dots N$  do
4:      $S \leftarrow$  rand( $1 \dots 100$ );
5:     if  $S \leq CR \parallel$  jrand =  $j$  then
6:        $X_{REC}^j \leftarrow X_{MUT}^j$ 
7:     else
8:        $X_{REC}^j \leftarrow X_i^j$ 
9:     end if
10:    return  $X_{REC}$ 
11:  end for
12: end function

```

Na recombinação binomial é verificado, para cada elemento do vetor paramétrico, qual será a origem do valor presente na solução recombinada (x_{REC}). Para isso, em cada posição é feito um sorteio e, com probabilidade CR , o elemento virá do vetor mutado (x_{MUT}), tendo em caso contrário origem no vetor de comparação na geração anterior (x_i). É ainda determinada uma posição aleatória do vetor (jrand) que possuirá, independente do sorteio, o valor correspondente do vetor mutado de forma a evitar que o vetor recombinado seja idêntico ao vetor de

comparação.

Algorithm 4 Recombinação Exponencial

```
1: function DE( $X_i$ ,  $X_{MUT}$ ,  $CR$ )
2:    $C \leftarrow 0$ ;
3:    $j \leftarrow \text{rand}(0 \dots N - 1)$ ; /* N dimensão do vetor paramétrico */
4:    $X_{REC} \leftarrow X_i$ 
5:   repeat
6:      $X_{REC}^{j+1} \leftarrow X_{MUT}^{j+1}$ ;
7:      $j \leftarrow (j + 1) \text{MOD } N$ ;
8:      $C \leftarrow C + 1$ ;
9:      $S \leftarrow \text{rand}(1 \dots 100)$ ;
10:  until  $S > CR \parallel C = N$ 
11:  return  $X_{REC}$ 
12: end function
```

No processo de recombinação exponencial, por outro lado, busca-se determinar uma região contígua do vetor paramétrico que receberá o valor do vetor mutado (x_{MUT}). Para isso é determinada aleatoriamente uma posição de início no vetor (j), sendo executados sucessivamente sorteios, tendo, para cada sorteio bem sucedido (probabilidade CR), o incremento de um elemento na região de troca. Este incremento é circular, dado que, ao se atingir o fim do vetor, o sorteio prosseguirá a partir do início do mesmo. O processo de incremento é finalizado quando há insucesso na sequência de sorteios (condição $S > CR$) ou quando todas as posições tenham sido selecionadas para troca.

3.5 Avaliação

A fase de avaliação implementa o processo de pressão seletiva, o que pode ser considerado o principal mecanismo presente em estratégias evolutivas. No contexto da metaheurística DE, a definição deste processo se dá com o estabelecimento de uma métrica, denominada função de aptidão, que torna possível quantificar a qualidade de uma dada solução da população e, conseqüentemente, compará-la a outros indivíduos.

Uma característica particular na DE, quando comparado a outras metaheurísticas baseadas em estratégias evolutivas, é que seu o método de avaliação tradi-

onalmente não verifica a população como um todo. Ao contrário, a implementação tradicional da DE avalia os indivíduos da população por comparação dois a dois, um destes pertencente a população existente na geração anterior e outro criado pelas operações de mutação e recombinação representando o novo conjunto de indivíduos, com o mais apto dentre estes dois mantido para a próxima geração. Isto faz com que cada elemento da população tenha chance de competir para permanecer na população, proporcionando maior variabilidade.

A fase de avaliação se dá por sucessivas comparações dois a dois entre os elementos até que todos os indivíduos tenham sido verificados uma única vez, tendo então o fim do processo em uma dada geração.

3.6 Tratamento de Restrições

A implementação geral da técnica DE, como é comum em outras metaheurísticas, não possui nenhum mecanismo para lidar com restrições funcionais de um problema. Desta forma, quando estas restrições estão presentes, se faz necessária a utilização de algum mecanismo extra a ser incluído na técnica.

Neste sentido, as abordagens para o tratamento de restrições podem ser classificadas em três mecanismos: estabelecimento de operações no algoritmo que mantenham sempre a viabilidade das soluções; correção ativa de soluções inviáveis de forma a reconduzi-las à região viável sempre que for verificada violação de alguma restrição; penalização das violações existentes de forma a refletir na qualidade do indivíduo sua inviabilidade.

Destas três opções, a manutenção por operações na região viável e correção da solução são opções muito dependentes do contexto da aplicação, apresentando complicações ao serem submetidas à restrições funcionais complexas ou de forma analítica desconhecida.

A penalização das violações, por outro lado, não apresenta estas dificuldades dado que tem como foco a situação momentânea de uma solução, necessitando apenas da violação de cada restrição para agir. Isto faz com que seja a opção mais

prática quando não há informação acerca da forma da restrição, o que, *a priori*, inviabilizaria as outras duas possibilidades. Apesar disto, a inclusão de um valor de penalização à avaliação de uma solução pode não ser tarefa simples e afetar a convergência de toda a técnica.

Capítulo 4

Algoritmo DE+Lemke

Tendo em vista a possibilidade de reescrita do BLPP como um problema de otimização com restrições lineares de complementaridade (Equação 2.10), a técnica DE+Lemke aqui proposta busca a utilização do método exato proposto por Lemke (1965) para resolver o subproblema de complementaridade linear e, de posse dos valores exatos de resposta do seguidor, resolver via DE o problema de otimização presente no líder.

Desta forma, pode-se dizer que a abordagem DE+Lemke é uma técnica de resolução aninhada para o BLPP, que combina programação matemática e metaheurística em sua execução, resolvendo o seguidor através do algoritmo de Lemke e o líder via evolução diferencial. Esta utilização de técnicas de programação matemática em implementações heurísticas está inclusa no contexto das chamadas *matheuristics*, área que vem crescendo em popularidade na comunidade acadêmica (Boschetti et al., 2009). Mais detalhes acerca da abordagem proposta serão apresentados nas próximas seções.

4.1 Estrutura

Como comentado, o algoritmo DE+Lemke busca aproveitar a resolução exata de MPECs proporcionada pela técnica Lemke, de forma a garantir correteza na resposta do seguidor para qualquer configuração do líder e, por consequência, na própria avaliação do líder.

O formato desta abordagem leva à utilização natural de uma técnica de resolução aninhada no BLPP, dado que o MPEC se refere somente às variáveis do seguidor, enquanto o líder necessita de uma outra maneira de otimizar suas próprias variáveis, no caso a DE. O Algoritmo 5 apresenta o pseudocódigo da técnica proposta.

Algorithm 5 DE+Lemke

```

1: function DE+LEMKE( $F, CR, NP$ )
2:    $X \leftarrow$  InicializaPopulaçãoAleatoriamente( $NP$ );
3:   for  $i \leftarrow 1 \dots NP$  do
4:      $y_{i,0} \leftarrow$  Lemke( $x_{i,0}$ )
5:   end for
6:    $G \leftarrow 0$ ;
7:   while Critério de parada do
8:      $G \leftarrow G + 1$ ;
9:     for  $i \leftarrow 1 \dots NP$  do
10:       $\{x_{r_1,G}, x_{r_2,G}, x_{r_3,G}\} \leftarrow$  SelecionaIndivíduos( $X_G$ ); /* $r_1 \neq r_2 \neq r_3$ */
11:       $x_{MUT} \leftarrow x_{r_1,G} + F(x_{best,G} - x_{r_1,G}) + F(x_{r_2,G} - x_{r_3,G})$ ;
12:       $jrand \leftarrow$  rand( $1 \dots N$ ); /*  $N$  dimensão do vetor paramétrico */
13:      for  $j \leftarrow 1 \dots N$  do
14:         $S \leftarrow$  rand( $1 \dots 100$ );
15:        if  $S \leq CR \parallel jrand = j$  then
16:           $x_{REC}^j \leftarrow x_{MUT}^j$ 
17:        else
18:           $x_{REC}^j \leftarrow x_{i,G}^j$ 
19:        end if
20:      end for
21:       $y_{REC} \leftarrow$  Lemke( $x_{REC}$ )
22:      if  $f_{DEB}(x_{REC}, y_{REC}) \leq f_{DEB}(x_{i,G}, y_{i,G})$  then
23:         $x_{i,G+1} \leftarrow x_{REC}$ 
24:      else
25:         $x_{i,G+1} \leftarrow x_{i,G}$ 
26:      end if
27:    end for
28:  end while
29:  return melhor avaliação para o líder em  $X$ 
30: end function

```

Considerando esta especificação, pode-se determinar os seguintes passos para execução da técnica DE+LEMKE.

Passo 1: A população é inicializada aleatoriamente, atribuindo valores às variáveis do líder dentro do intervalo determinado por seus limitantes. Para estas soluções iniciais é determinado, via método de Lemke, o valor para as variáveis do

seguidor.

Passo 2: Para cada elemento da nova população $G + 1$ são selecionados 3 indivíduos da geração atual G , sendo estes utilizados na aplicação do operador de mutação rand-to-best/1 dando origem à solução x_{MUT} .

Passo 3: A seguir, aplica-se a técnica de cruzamento binomial combinando os elementos do vetor mutado x_{MUT} e do indivíduo da geração atual x_i , criando um novo elemento x_{REC} . A este, o método de Lemke atribui os valores exatos para as variáveis do seguidor.

Passo 4: Por fim, o novo elemento x_{REC} é avaliado e comparado com o elemento atual da população x_i , sendo o melhor entre estes levado à população em $G + 1$.

Passo 5: O processo é finalizado através de um critério de parada especificado. Caso este não tenha sido atendido, retorna ao Passo 2.

Uma discussão mais detalhada deste processo é apresentada a seguir, tendo destaque as adaptações realizadas.

4.2 Otimização no nível do líder via DE

A metaheurística evolução diferencial é utilizada no nível do líder para buscar a configuração que forneça seu valor ótimo. Sua implementação é feita de forma que altere apenas as variáveis do nível do líder, não estando qualquer elemento do seguidor submetido às suas operações.

Na definição da abordagem foi selecionada a variante DE/rand-to-best/1/bin que teve o melhor desempenho entre todas as combinações de operações verificadas. Ainda, outras definições importantes podem ser citadas acerca da DE.

Como primeiro ponto, a população inicial é gerada aleatoriamente, através do sorteio de valores dentro dos limitantes fornecidos pela instância para cada variável (ou um número arbitrariamente grande quando esta for ilimitada).

O operador de mutação implementado não utiliza controle de viabilidade para as restrições de “caixa”, prática comum em implementações DE, dado que

foi observado um melhor desempenho ao se optar por uma maior liberdade de movimentos. Desta forma, a busca pela manutenção das soluções na “caixa” viável ocorre no processo de avaliação, seguindo o mesmo procedimento das restrições funcionais do problema.

No cruzamento não há qualquer diferenciação das implementações clássicas da técnica binomial, seguindo o mesmo formato visto no Capítulo 3.

Por fim, a avaliação da qualidade de uma solução é feita através da aplicação do critério de Deb, que foi submetido a pequenas alterações para adequá-lo ao contexto do problema de programação em dois níveis. Maiores detalhes deste tópico podem ser vistos na Subseção 4.2.1.

4.2.1 Método de Deb aplicado ao DE+LEMKE

Para o tratamento de restrições foi utilizado o esquema de seleção proposto por Deb (Deb, 2000).

Neste método uma solução viável é sempre avaliada como de melhor qualidade se comparada a uma solução inviável, independentemente do valor da função objetivo. Para casos em que ambas as soluções comparadas são inviáveis, é feito o somatório das violações nas restrições, sendo a solução de menor valor nessa medida tomada como de melhor avaliação. Por fim, para casos de duas soluções viáveis, é verificado o valor de função objetivo de ambas; com a de menor valor na função (para o caso de minimização) sendo avaliada como melhor.

Considerando o caso específico do BLPP foi ainda adicionado um outro nível de seleção, relativo às soluções em que não foi possível determinar valores para a resposta do seguidor. Estas soluções, denominadas aqui incompletas, foram consideradas como de mais baixo nível de prioridade sendo preteridas por quaisquer soluções completas, sejam estas viáveis ou não.

Para comparação entre duas soluções incompletas, é verificada uma possível violação quanto às restrições de “caixa” para variáveis no nível do líder, com a de menor somatório das violações mantida na população, esquema análogo ao de

soluções inviáveis no nível do líder.

Nota-se que não é necessário verificar violações relativas às restrições presentes no seguidor, dado que este é resolvido exatamente (quando possível). Desta forma, a solução retornada pelo método de Lemke será necessariamente viável no nível do seguidor, tendo apenas que ser verificadas as restrições relativas ao nível do líder.

Um ponto ainda a ser comentado, é que, no sentido de economizar recursos, é feita a verificação do valor da função objetivo apenas nos casos em que a solução se apresente viável, tendo em vista que, pelo critério Deb, não é utilizada informação de função objetivo na avaliação de soluções inviáveis.

4.3 Resolução via Lemke para o seguidor

O algoritmo de Lemke foi implementado da forma apresentada na Subseção 2.2.1 deste trabalho, sendo utilizado para determinar a resposta do seguidor para uma dada configuração do líder.

O algoritmo de Lemke recebe as soluções geradas pela DE no nível do líder, sendo executado até sua condição de parada ou atingir um número de operações de pivotamento maior que o limite estipulado na implementação (arbitrariamente 100 operações de pivotamento). O fim da execução se dá com o método fornecendo os valores encontrados para o seguidor, sendo estes atribuídos à solução, ou informando que, para a configuração atribuída ao líder, não foi possível obter valores válidos para o MPEC, sendo atribuída à solução o rótulo de solução incompleta.

Ainda, algumas dificuldades podem ocorrer em configurações que levam à existência de múltiplas soluções no seguidor, não sendo possível, *a priori*, definir, via algoritmo de Lemke, uma estratégia otimista ou pessimista, como comentado anteriormente no Capítulo 2.

De qualquer forma foi verificado, para todos os casos de teste utilizados neste trabalho, a convergência do método de Lemke ao menos no ponto ótimo conhecido, bem como a capacidade de convergência da técnica DE+LEMKE a este ponto, o que foi considerado suficiente para os propósitos aqui estipulados.

Capítulo 5

Casos de Teste

Para validação das técnicas desenvolvidas e verificação de sua acurácia e robustez foram selecionados diversos casos de teste. Estes podem ser agrupados em dois tipos: casos de teste da literatura e casos de teste escaláveis.

Os aqui denominados casos de teste da literatura envolvem instâncias comumente utilizadas em artigos vinculados ao BLPP, com poucas variáveis e não-escaláveis.

O segundo conjunto, denominado casos de teste escaláveis, está vinculado a casos de teste desenvolvidos de forma que se possa expandir o número de variáveis e/ou restrições tornando-os gradualmente mais complexos. Mais detalhes de ambas as classes são apresentados nas próximas seções.

5.1 Instâncias-Teste na Literatura

Casos de teste presentes na literatura são de importante utilização por permitirem verificar o desempenho das técnicas desenvolvidas possibilitando, devido à larga utilização destas instâncias, a comparação deste trabalho com outros já presentes na literatura. São instâncias, em geral, de poucas variáveis, ajustadas manualmente de forma a apresentarem dificuldades específicas a serem resolvidas. Estas dificuldades podem ser vinculadas à forma das funções objetivo e das restrições, interatividade entre líder e seguidor, entre outros fatores.

Neste contexto, uma questão a ser resolvida em algumas instâncias é a defi-

nição de limites superiores e/ou inferiores para variáveis em casos onde estes não estão presentes. Este fator é especialmente sensível em metaheurísticas populacionais, onde é necessária a utilização de um grande número de pontos selecionados aleatoriamente em uma região do espaço, tendo esta que ser necessariamente limitada. Um recurso possível seria a definição de limites arbitrariamente grandes, mas estes criam outro obstáculo quando as próprias restrições do problema impõem valores máximos ou mínimos implicitamente, gerando complexidade desproporcional na viabilização das soluções.

A dificuldade ainda se acentua no BLPP, onde soluções que violem os limitantes induzidos pelas restrições do nível do seguidor são necessariamente soluções incompletas. Em soluções incompletas, como não se sabe o comportamento esperado para o seguidor, *a priori* não é possível definir quaisquer valores de violação para as restrições ou de função objetivo, fazendo com que a solução em si não contribua com qualquer informação ao algoritmo de resolução. Caso ainda a região inviável no nível do seguidor represente uma grande porção do espaço de busca, o número de soluções incompletas pode ser demasiadamente grande, o que reduziria a execução de um método como a evolução diferencial a uma busca quase cega, inviabilizando qualquer análise de desempenho.

Como não é do escopo deste trabalho lidar com tais dificuldades, foram determinados, a partir das restrições do problema, valores máximos e mínimos possíveis para as variáveis que não foram estipulados na formulação original.

Foram selecionados 18 casos de teste restritos a funções lineares e quadráticas e restrições lineares no nível do seguidor. Estes foram escolhidos considerando especialmente a dificuldade de convergência, de forma a uma melhor análise de desempenho da técnica aqui proposta, bem como características vistas como pertinentes na formulação de cada instância. Estas são apresentadas a seguir na Tabela 5.1. Nesta tabela é possível verificar a forma da função analisada, a referência de onde foi obtida (coluna Casos de Teste) e o melhor valor conhecido na literatura para a função objetivo do líder com respectiva avaliação para o seguidor (coluna

Resultado).

Os valores na coluna resultado foram obtidos respectivamente nos trabalhos Angelo et al. (2013) [Pr1-Pr8], Li e Zhang (2014) [Pr9-Pr15] e Li e Wang (2006) [Pr16-Pr18].

Casos de Teste	Resultado
<p>Pr1</p> $\min_x F(x, y) = (x_1 - 30)^2 + (x_2 - 20)^2 - 20y_1 + 20y_2$ <p>s.t. $x_1 + 2x_2 \geq 30$</p> $x_1 + x_2 \leq 25$ $x_2 \leq 15$ $\min_y f(x, y) = (x_1 - y_1)^2 + (x_2 - y_2)^2$ <p>s.t. $y_1, y_2 \in [0, 10]$</p> <p>Shimizu e Aiyoshi (1981)</p>	$F(x, y) = 225$ $f(x, y) = 100$
<p>Pr2</p> $\max_x F(x, y) = 2x_1 - x_2 - 0.5y_1$ <p>s.t. $\max_y f(x, y) = -x_1 - x_2 + 4y_1 - y_2$</p> <p>s.t. $2x_1 - y_1 + y_2 \geq 2.5$</p> $-x_1 + 3x_2 - y_2 \geq -2$ $-x_1 - x_2 \geq -2$ $x, y \geq 0$ <p>Bard e Falk (1982)</p>	$F(x, y) = 3.25$ $f(x, y) = 4$
<p>Pr3</p> $\min_x F(x, y) = 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60$ <p>s.t. $x_1 + x_2 + y_1 - 2y_2 - 40 \leq 0$</p> $\min_y f(x, y) = (y_1 - x_1 + 20)^2 + (y_2 - x_2 + 20)^2$ <p>s.t. $2y_1 - x_1 + 10 \leq 0$</p> $2y_2 - x_2 + 10 \leq 0$ $x \in [0, 50], y \in [-10, 20]$ <p>Aiyoshi e Shimizu (1984)</p>	$F(x, y) = 0$ $f(x, y) = 200$

Pr4	$\min_x F(x, y) = (x - 5)^2 + (2y + 1)^2$ $\text{s.t.} \quad \min_y f(x, y) = (y - 1)^2 - 1.5xy$ $\text{s.t.} \quad 3x - y \geq 3$ $-x + 0.5y \geq -4$ $-x - y \geq -7$ $x, y \geq 0$ <p style="text-align: center;">Bard (1988)</p>	$F(x, y) = 17$ $f(x, y) = 1$
Pr5	$\min_x F(x, y) = -x_1^2 - 3x_2 - 4y_1 + y_2^2$ $\text{s.t.} \quad x_1^2 + 2x_2 \leq 4$ $\min_y f(x, y) = 2x_1^2 + y_1^2 - 5y_2$ $\text{s.t.} \quad x_1^2 - 2x_1 + x_2^2 - 2y_1 + y_2 \geq -3$ $x_2 + 3y_1 - 4y_2 \geq 4$ $x, y \geq 0$ <p style="text-align: center;">Bard (1988)</p>	$F(x, y) = -12.679$ $f(x, y) = -1.0156$
Pr6	$\min_x F(x, y) = (x - 1)^2 + 2y_1 - 2x$ $\text{s.t.} \quad \min_y f(x, y) = (2y_1 - 4)^2 + (2y_2 - 1)^2 + xy_1$ $\text{s.t.} \quad 4x + 5y_1 + 4y_2 \leq 12$ $-4x - 5y_1 + 4y_2 \leq -4$ $4x - 4y_1 + 5y_2 \leq 4$ $-4x + 4y_1 + 5y_2 \leq 4$ $x, y \geq 0$ <p style="text-align: center;">Savard e Gauvin (1994)</p>	$F(x, y) = -1.21$ $f(x, y) = 7.61$
Pr7	$\min_x F(x, y) = (x - 1)^2 + (y - 1)^2$ $\text{s.t.} \quad \min_y f(x, y) = 0.5y^2 + 500y - 50xy$ $x, y \geq 0$ <p style="text-align: center;">Bard (1998)</p>	$F(x, y) = 1$ $f(x, y) = 0$

Pr8	$\min_x F(x, y) = (x - 3)^2 + (y - 2)^2$ $\text{s.t.} \quad \min_y f(x, y) = (y - 5)^2$ $\text{s.t.} \quad 2x - y \geq -1$ $-x + 2y \geq 2$ $-x - 2y \geq -14$ $x \in [0, 8], y \geq 0$ Rajesh et al. (2003)	$F(x, y) = 5$ $f(x, y) = 4$
Pr9	$\min_x F(x, y) = -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3$ $\text{s.t.} \quad x_1, x_2 \geq 0$ $\min_y f(x, y) = x_1 + 2x_2 + y_1 + y_2 + 2y_3$ $\text{s.t.} \quad -y_1 + y_2 + y_3 \leq 1$ $2x_1 - y_1 + 2y_2 - 0.5y_3 \leq 1$ $2x_2 + 2y_1 - y_2 - 0.5y_3 \leq 1$ $y_1, y_2, y_3 \geq 0$ Bard (1998)	$F(x, y) = -29.2$ $f(x, y) = 3.2$
Pr10	$\min_x F(x, y) = -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3$ $\text{s.t.} \quad x_1 + 2x_2 - y_3 \leq 1.3$ $x_1, x_2 \geq 0$ $\min_y f(x, y) = 2y_1 + y_2 + 2y_3$ $\text{s.t.} \quad -y_1 + y_2 + y_3 \leq 1$ $4x_1 - 2y_1 + 4y_2 - y_3 \leq 2$ $4x_2 + 4y_1 - 2y_2 - y_3 \leq 2$ $y_1, y_2, y_3 \geq 0$ Hansen et al. (1992)	$F(x, y) = -18.4$ $f(x, y) = 1.8$

Pr11

$$\min_x F(x, y) = -4x_1 + 8x_2 + x_3 - x_4 + 9y_1 - 9y_2$$

$$\text{s.t.} \quad -9x_1 + 3x_2 - 8x_3 + 3x_4 + 3y_1 \leq 1$$

$$4x_1 - 10x_2 + 3x_3 + 5x_4 + 8y_1 + 8y_2 \leq 25$$

$$4x_1 - 2x_2 - 2x_3 + 10x_4 - 5y_1 + 8y_2 \leq 21$$

$$9x_1 - 9x_2 + 4x_3 - 3x_4 - y_1 - 9y_2 \leq -1$$

$$-2x_1 - 2x_2 + 8x_3 - 5x_4 + 5y_1 + 8y_2 \leq 20$$

$$7x_1 + 2x_2 - 5x_3 + 4x_4 - 5y_1 \leq 11$$

$$x_1, x_2, x_3, x_4 \geq 0$$

$$\min_y f(x, y) = -9y_1 + 9y_2$$

$$\text{s.t.} \quad -6x_1 + x_2 + x_3 - 3x_4 - 9y_1 - 7y_2 \leq -15$$

$$4x_2 + 5x_3 + 10x_4 \leq 26$$

$$-9x_1 + 9x_2 - 9x_3 + 5x_4 - 5y_1 - 4y_2 \leq -5$$

$$5x_1 + 3x_2 + x_3 + 9x_4 + y_1 + 5y_2 \leq 32$$

$$y_1, y_2 \geq 0$$

Moshirvaziri et al. (1996)

$$F(x, y) = 14.99$$

$$f(x, y) = -16.99$$

Pr12

$$\min_x F(x, y) = 12x_1 - x_2 - 12x_3 + 13x_4 + 2x_6 - 5x_8 + 6x_9$$

$$-11x_{10} - 5y_1 - 6y_2 - 4y_3 - 7y_4$$

$$\text{s.t. } Ax + By \leq r_1$$

$$0 \leq x \leq 10$$

$$\min_y f(x, y) = 3y_1 - 2y_2 - 3y_3 - 3y_4 + y_5 + 6y_6$$

$$\text{s.t. } Cx + Dy \leq r_2$$

$$0 \leq y \leq 10$$

where:

$$A = \begin{bmatrix} -2 & -3 & 14 & -2 & -9 & 2 & 1 & -4 & 0 & 2 \\ 1 & -7 & 13 & 0 & -15 & 2 & -8 & -4 & 4 & -7 \end{bmatrix},$$

$$B = \begin{bmatrix} -3 & 9 & -2 & -8 & 1 & -8 \\ -6 & -2 & 6 & 2 & 8 & -4 \end{bmatrix},$$

$$C = \begin{bmatrix} -5 & 7 & 4 & -2 & 3 & -9 & 9 & -1 & -3 & 11 \\ 6 & -5 & -3 & -2 & 8 & 5 & 8 & -3 & 7 & 3 \\ -6 & -4 & 2 & 0 & -2 & 3 & -3 & 2 & 2 & 4 \\ 5 & 6 & 0 & -4 & 3 & -8 & 1 & 0 & 2 & -3 \\ 11 & -11 & 4 & 5 & -10 & -6 & 14 & -7 & -11 & -3 \\ 9 & -12 & -4 & -10 & 2 & 8 & 5 & -11 & -4 & 1 \\ 7 & -2 & -6 & 0 & -11 & 1 & -2 & -2 & -1 & -2 \end{bmatrix},$$

$$D = \begin{bmatrix} 10 & -9 & -6 & 4 & 6 & -3 \\ -5 & -7 & 1 & 1 & -6 & 4 \\ 10 & 5 & 6 & -4 & 3 & -1 \\ -4 & -3 & -4 & -4 & 1 & 1 \\ -10 & -7 & 7 & 7 & 2 & 7 \\ 2 & -5 & 10 & 1 & 4 & 5 \\ -5 & -5 & -6 & -5 & 1 & -12 \end{bmatrix},$$

$$r_1 = (30, -134)^T, r_2 = (83, 92, 168, -96, -133, 89, -192)^T$$

Tuy et al. (2007)

$$F(x, y) = -453.61$$

$$f(x, y) = -68.81$$

Pr13

$$\max_x F(x, y) = (y_1 + y_3)(200 - y_1 - y_3) +$$

$$(y_2 + y_4)(160 - y_2 - y_4)$$

$$\text{s.t. } x_1 + x_2 + x_3 + x_4 \leq 40$$

$$0 \leq x_1 \leq 10, 0 \leq x_2 \leq 5,$$

$$0 \leq x_3 \leq 15, 0 \leq x_4 \leq 20$$

$$\min_y f_1(x, y) = (y_1 - 4)^2 + (y_2 - 13)^2$$

$$\text{s.t. } 0.4y_1 + 0.7y_2 - x_1 \leq 0$$

$$0.6y_1 + 0.3y_2 - x_2 \leq 0$$

$$y \in [0, 20]$$

$$\min_y f_2(x, y) = (y_3 - 35)^2 + (y_4 - 2)^2$$

$$\text{s.t. } 0.4y_3 + 0.7y_4 - x_3 \leq 0$$

$$0.6y_3 + 0.3y_4 - x_4 \leq 0$$

$$y \in [0, 40]$$

Bard (1988)

$$F(x, y) = 6600$$

$$f_1(x, y) = 24.58$$

$$f_2(x, y) = 29.54$$

Pr14

$$\min_x F(x, y) = |2x_1 + 2x_2 - 3y_1 - 3y_2 - 60|$$

$$\text{s.t. } x_1 + x_2 + y_1 - 2y_2 \leq 40$$

$$x \in [0, 50]$$

$$\min_y f(x, y) = (y_1 - x_1 + 20)^2 + (y_2 - x_2 + 20)^2$$

$$\text{s.t. } 2y_1 - x_1 + 10 \leq 0$$

$$2y_2 - x_2 + 10 \leq 0$$

$$y \in [-10, 20]$$

Wang et al. (2005)

$$F(x, y) = 0$$

$$f(x, y) = 100$$

Pr15	$\min_x F(x, y) = (x - 1)^2 + 2y_1 - 2x + 1.2097 $ $\text{s.t. } x \geq 0$ $\min_y f(x, y) = (2y_1 - 4)^2 + (2y_2 - 1)^2 + xy_1$ $\text{s.t. } 4x + 5y_1 + 4y_2 \leq 12$ $4y_2 - 4x - 5y_1 \leq -4$ $4x - 4y_1 + 5y_2 \leq 4$ $4y_1 - 4x + 5y_2 \leq 4$ $y \geq 0$ <p style="text-align: center;">Wang et al. (2005)</p>	$F(x, y) = 0$ $f(x, y) = 7.61$
Pr16	$\min_x F(x, y) = 0.1(x_1^2 + x_2^2) - 3y_1 - 4y_2 + 0.5(y_1^2 + y_2^2)$ $\text{s.t. } \min_y f(x, y) = 0.5(y_1^2 + 6y_1y_2 + 10y_2^2) -$ $y_1(2x_2 - x_1) - y_2(3x_1 - 3x_2)$ $\text{s.t. } -0.333y_1 + y_2 - 2 \leq 0$ $y_1 - 0.333y_2 - 2 \leq 0$ $y \geq 0$ <p style="text-align: center;">Outrata (1990)</p>	$F(x, y) = -3.92$ $f(x, y) = -2$
Pr17	$\min_x F(x, y) = 0.5((y_1 - 3)^2 + (y_2 - 4)^2)$ $\text{s.t. } \min_y f(x, y) = 0.5(y_1^2 + y_2^2) - y_1(3 + 1.333x) - y_2x$ $\text{s.t. } (-0.333 + 0.1x)y_1 + y_2 \leq x$ $y_1 + (-0.333 - 0.1x)y_2 \leq 2$ $x, y \geq 0$ <p style="text-align: center;">Outrata (1990)</p>	$F(x, y) = 0.8485$ $f(x, y) = -22.951$
Pr18	$\min_x F(x, y) = 0.5((y_1 - 3)^2 + (y_2 - 4)^2)$ $\text{s.t. } \min_y f(x, y) = 0.5(y_1^2(1 + 0.2x) + y_2^2(1 + 0.1x)) -$ $y_1(3 + 1.333x) - y_2x$ $\text{s.t. } (-0.333 + 0.1x)y_1 + y_2 \leq 2 - 0.1x$ $y_1 + (-0.333 - 0.1x)y_2 \leq 2 - 0.1x$ $x, y \geq 0$ <p style="text-align: center;">Outrata (1990)</p>	$F(x, y) = 1.5629$ $f(x, y) = -11.683$

Tabela 5.1: Casos de Teste da Literatura

5.2 Instâncias-Teste Escaláveis

Uma deficiência presente nos casos de teste da literatura para o BLPP é a relativa facilidade que muitos destes apresentam em sua resolução. Esta questão pode ser vinculada em parte ao grau de dificuldade do BLPP em si e em parte a muitas destas instâncias terem sido elaboradas há vários anos, tendo sido desenvolvidas em outro contexto de técnicas e poder computacional. Desta forma, apesar de possuírem utilidade na validação de técnicas, não apresentam, em determinados casos, desafio suficiente para identificação de limitações das mesmas.

Neste sentido, uma boa iniciativa pode ser vista no trabalho de Sinha et al. (2013), que desenvolveu uma metodologia para a criação de instâncias escaláveis, além de propor um conjunto próprio de instâncias criadas desta forma. Estas instâncias, conhecidas como SMD, têm suas dimensões definidas pelos parâmetros p, q, r, s , com cada um destes incrementando a dimensão do problema e aplicando funcional específico a estas novas dimensões. O conjunto apresentado pelo trabalho de Sinha et al. (2013) é composto por 8 instâncias sem restrições funcionais (SMD1, SMD2, SMD3, SMD4, SMD5, SMD6, SMD7, SMD8) e 4 instâncias incluindo tanto restrições de “caixa” como restrições funcionais (SMD9, SMD10, SMD11, SMD12).

Apesar destes pontos positivos, as instâncias propostas em Sinha et al. (2013) foram desenvolvidas considerando a formulação geral do BLPP, o que impede que sejam, em sua maior parte, utilizadas para testar a proposta deste trabalho que supõe que a função objetivo do seguidor seja quadrática nas variáveis do seguidor e com restrições lineares. Para contornar este problema e inspirando-se no trabalho de Sinha et al. (2013), foram propostas aqui novas instâncias geradas via duas metodologias.

Em um primeiro momento, para instâncias sem restrições, foram utilizadas as funções SMD1, SMD2, SMD3, SMD4 e SMD5 de Sinha et al. (2013) que foram adaptadas de forma a se enquadrarem às condições aqui impostas formando o conjunto de instâncias denominado aqui SMDquad. A instância SMD6 foi incluída sem modificações a este conjunto, uma vez atende às condições estabelecidas neste

trabalho, enquanto que SMD7 e SMD8 foram omitidas.

Para instâncias com restrições, foram criados casos de teste, independentemente, utilizando apenas a metodologia proposta em Sinha et al. (2013). Estas instâncias, denominadas Cquad, apresentam restrições desenvolvidas visando dificultar a convergência dos métodos de resolução, permitindo que tanto o número de variáveis quanto o de restrições possam ser escaláveis. A seguir serão apresentados mais detalhes de cada uma das funções desenvolvidas.

5.2.1 Instâncias-Teste sem Restrições Funcionais

As Tabelas 5.2, 5.3, 5.4, 5.5 e 5.6 apresentam os casos de teste SMDquad1, SMDquad2, SMDquad3, SMDquad4 e SMDquad5 do grupo SMDquad comparando estes às respectivas versões originais em SMD. Nestas instâncias as variáveis para líder e seguidor estão separadas em duas componentes: x_1 e x_2 para o líder e y_1 e y_2 para o seguidor, sendo a componente x_1 atrelada aos funcionais F_1 e f_1 , a componente y_1 atrelada aos funcionais F_2 e f_2 e as componentes x_2 e y_2 atreladas aos funcionais F_3 e f_3 . Para os casos SMD1-SMD5 e versões modificadas, a dimensão do nível do líder se dá pela soma dos parâmetros p e r , sendo a dimensão do seguidor obtida pela soma de q e r .

A principal adaptação feita entre as duas versões de instâncias, SMD e SMDquad, foi a alteração da função F_3 e sua correspondente no seguidor, f_3 , trocando-se o funcional aplicado às variáveis do nível do seguidor com as do líder, de forma a tornar a função objetivo do seguidor quadrática. Ainda foram feitas, nas funções em que era necessário, alterações dos funcionais F_2 e f_2 , buscando aproximá-los, em alguma medida, do comportamento original das instâncias SMD.

Mais detalhadamente, o caso de teste SMDquad1 troca o termo $(x_2^i - \tan(y_2^i))$ para $(y_2^i - \tan(x_2^i))$, buscando manter um formato semelhante na função objetivo do líder (considerando resposta correta do seguidor) e alterando apenas as restrições de “caixa”.

SMD1	SMDquad1
$\min_x F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$ $F_1(x_1) = \sum_{i=1}^p (x_1^i)^2$ $F_2(y_1) = \sum_{i=1}^q (y_1^i)^2$ $F_3(x_2, y_2) = \sum_{i=1}^r (x_2^i)^2 + \sum_{i=1}^r (x_2^i - \tan(y_2^i))^2$ s.t. $x_1 \in [-5, 10]$ $x_2 \in [-5, 10]$ $\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$ $f_1(x_1, x_2) = \sum_{i=1}^p (x_1^i)^2$ $f_2(y_1) = \sum_{i=1}^q (y_1^i)^2$ $f_3(x_2, y_2) = \sum_{i=1}^r (x_2^i - \tan(y_2^i))^2$ s.t. $y_1 \in [-5, 10]$ $y_2 \in (-\pi/2, \pi/2)$	$\min_x F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$ $F_1(x_1) = \sum_{i=1}^p (x_1^i)^2$ $F_2(y_1) = \sum_{i=1}^q (y_1^i)^2$ $F_3(x_2, y_2) = \sum_{i=1}^r (x_2^i)^2 + \sum_{i=1}^r (y_2^i - \tan(x_2^i))^2$ s.t. $x_1 \in [-5, 10]$ $x_2 \in (-\pi/2, \pi/2)$ $\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$ $f_1(x_1, x_2) = \sum_{i=1}^p (x_1^i)^2$ $f_2(y_1) = \sum_{i=1}^q (y_1^i)^2$ $f_3(x_2, y_2) = \sum_{i=1}^r (y_2^i - \tan(x_2^i))^2$ s.t. $y_1 \in [-5, 10]$ $y_2 \in [-5, 10]$

Tabela 5.2: SMD1 x SMDquad1

Nesta elaboração, a relação de otimalidade líder-seguidor é $y_1^i = 0$, $i = 1, \dots, q$, $y_2^i = \tan(x_2^i)$, $i = 1, \dots, r$, com ótimo global em $x_1^i = 0$, $i = 1, \dots, p$, $x_2^i = 0$, $i = 1, \dots, r$. O valor ótimo global da função objetivo do líder é $F(x, y) = 0$.

A Figura 5.1 mostra o comportamento da função objetivo do líder em relação às suas variáveis, assumindo resposta ótima do seguidor para todos os pontos desta instância e com parâmetros $p = 1$, $q = 1$ e $r = 1$ (como as componentes x_1 e x_2 tem apenas uma dimensão, seu índice é omitido no gráfico). É possível notar que, considerando o seguidor sempre com resposta correta, a função objetivo do líder se reduz a um parabolóide centrado em $(0, 0)$.

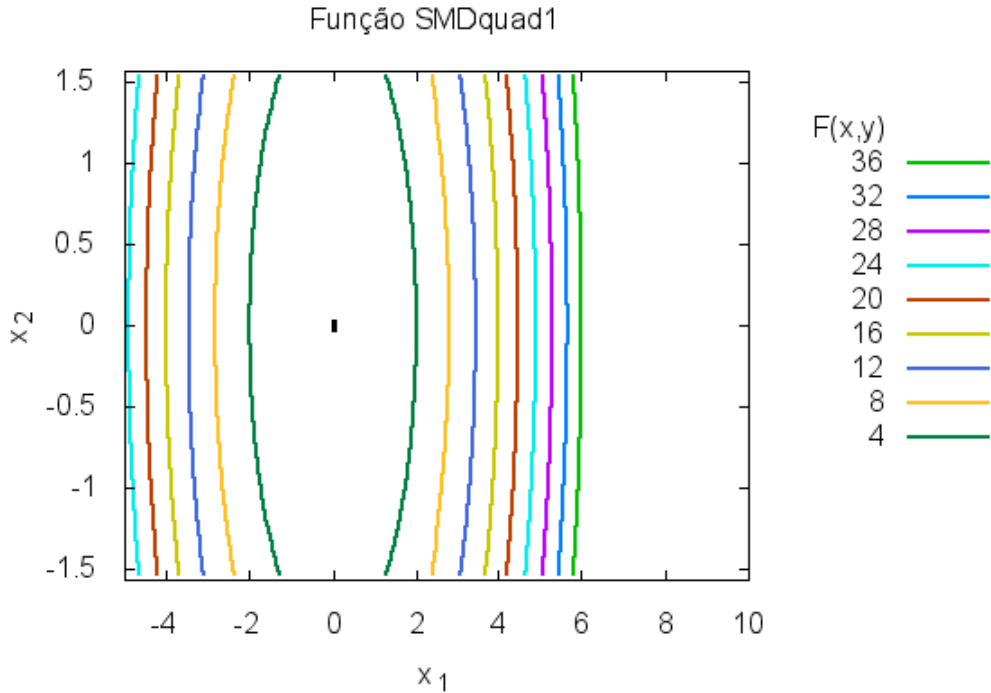


Figura 5.1: Função objetivo do Nível do Líder: SMDquad1

No caso de teste SMDquad2 a troca ocorre no termo $(x_2^i - \log(y_2^i))$ que foi alterado para $(y_2^i - \log(x_2^i))$, tendo um comportamento próximo novamente ao caso SMD (tendo em conta a correta resposta do seguidor) e alterando apenas as restrições de “caixa”.

SMD2	SMDquad2
$\min_x F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$ $F_1(x_1) = \sum_{i=1}^p (x_1^i)^2$ $F_2(y_1) = -\sum_{i=1}^q (y_1^i)^2$ $F_3(x_2, y_2) = \sum_{i=1}^r (x_2^i)^2 - \sum_{i=1}^r (x_2^i - \log(y_2^i))^2$ s.t. $x_1 \in [-5, 10]$ $x_2 \in [-5, 1]$ $\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$ $f_1(x_1, x_2) = \sum_{i=1}^p (x_1^i)^2$ $f_2(y_1) = \sum_{i=1}^q (y_1^i)^2$ $f_3(x_2, y_2) = \sum_{i=1}^r (x_2^i - \log(y_2^i))^2$ s.t. $y_1 \in [-5, 10]$ $y_2 \in (0, e]$	$\min_x F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$ $F_1(x_1) = \sum_{i=1}^p (x_1^i)^2$ $F_2(y_1) = -\sum_{i=1}^q (y_1^i)^2$ $F_3(x_2, y_2) = \sum_{i=1}^r (x_2^i - 1)^2 - \sum_{i=1}^r (y_2^i - \log(x_2^i))^2$ s.t. $x_1 \in [-5, 10]$ $x_2 \in [1/e, e]$ $\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$ $f_1(x_1, x_2) = \sum_{i=1}^p (x_1^i)^2$ $f_2(y_1) = \sum_{i=1}^q (y_1^i)^2$ $f_3(x_2, y_2) = \sum_{i=1}^r (y_2^i - \log(x_2^i))^2$ s.t. $y_1 \in [-5, 10]$ $y_2 \in [-5, 1]$

Tabela 5.3: SMD2 x SMDquad2

Agora a relação de otimalidade líder-seguidor é $y_1^i = 0$, $i = 1, \dots, q$, $y_2^i = \log(x_2^i)$, $i = 1, \dots, r$, com ótimo global em $x_1^i = 0$, $i = 1, \dots, p$, $x_2^i = 1$, $i = 1, \dots, r$. O valor ótimo global da função objetivo do líder é $F(x, y) = 0$.

A Figura 5.2 mostra o comportamento da função objetivo do líder em relação às suas variáveis, assumindo resposta ótima do seguidor para todos os pontos desta instância e com parâmetros $p = 1$, $q = 1$ e $r = 1$ (como as componentes x_1 e x_2 tem apenas uma dimensão, seu índice é omitido no gráfico). É possível notar que, considerando o seguidor sempre com resposta correta, a função objetivo do líder se reduz a um paraboloide centrado em $(0, 1)$.

No caso de teste SMDquad3 a troca ocorre no termo $((x_2^i)^2 - \tan(y_2^i))$ que foi alterado para $(y_2^i - \tan(\sqrt{x_2^i}))$, se aproximando novamente do formato da função objetivo do líder e alterando apenas as restrições de “caixa”. Modificou-se ainda a função f_2 , com o termo $q + \sum_{i=1}^q ((y_1^i)^2 - \cos(2\pi y_1^i))$ mudado para $q + \sum_{i=1}^q ((y_1^i)^2 - 1)$.

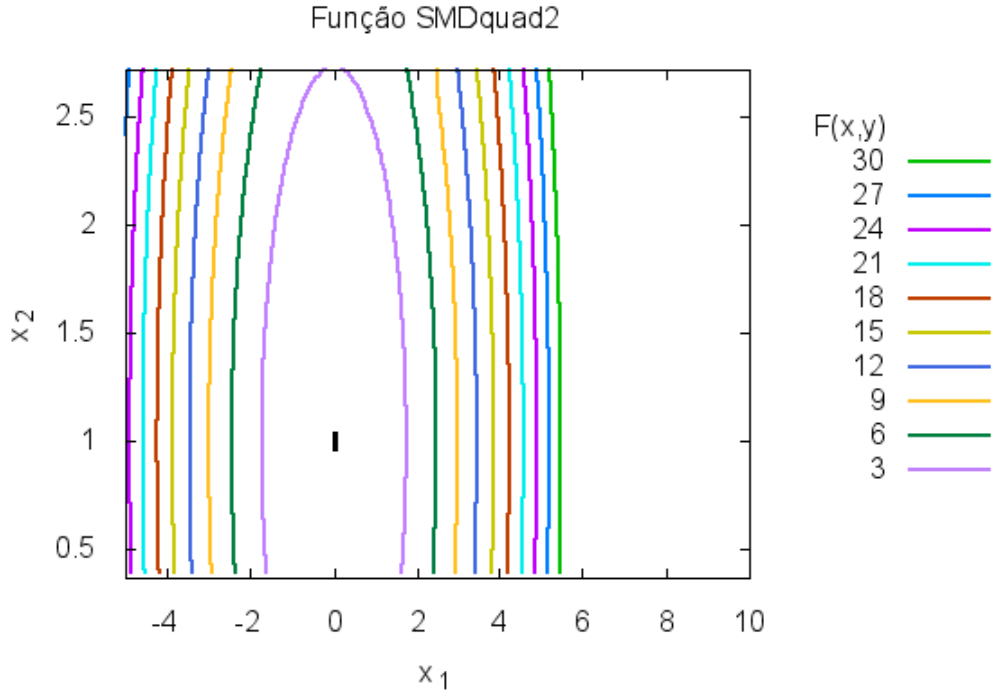


Figura 5.2: Função objetivo do Nível do Líder: SMDquad2

SMD3	SMDquad3
$\min_x F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$ $F_1(x_1) = \sum_{i=1}^p (x_1^i)^2$ $F_2(y_1) = \sum_{i=1}^q (y_1^i)^2$ $F_3(x_2, y_2) = \sum_{i=1}^r (x_2^i)^2 + \sum_{i=1}^r ((x_2^i)^2 - \tan(y_2^i))^2$ $\text{s.t. } x_1 \in [-5, 10]$ $x_2 \in [-5, 10]$ $\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$ $f_1(x_1, x_2) = \sum_{i=1}^p (x_1^i)^2$ $f_2(y_1) = q + \sum_{i=1}^q ((y_1^i)^2 - \cos(2\pi y_1^i))$ $f_3(x_2, y_2) = \sum_{i=1}^r ((x_2^i)^2 - \tan(y_2^i))^2$ $\text{s.t. } y_1 \in [-5, 10]$ $y_2 \in (-\pi/2, \pi/2)$	$\min_x F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$ $F_1(x_1) = \sum_{i=1}^p (x_1^i)^2$ $F_2(y_1) = \sum_{i=1}^q (y_1^i)^2$ $F_3(x_2, y_2) = \sum_{i=1}^r (x_2^i)^2 + \sum_{i=1}^r (y_2^i - \tan(\sqrt{x_2^i}))^2$ $\text{s.t. } x_1 \in [-5, 10]$ $x_2 \in [0, \pi/2)$ $\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$ $f_1(x_1, x_2) = \sum_{i=1}^p (x_1^i)^2$ $f_2(y_1) = q + \sum_{i=1}^q ((y_1^i)^2 - 1)$ $f_3(x_2, y_2) = \sum_{i=1}^r (y_2^i - \tan(\sqrt{x_2^i}))^2$ $\text{s.t. } y_1 \in [-5, 10]$ $y_2 \in [-5, 10]$

Tabela 5.4: SMD3 x SMDquad3

Nesta elaboração, a relação de otimalidade líder-seguidor é $y_1^i = 0, i = 1, \dots, q$, $y_2^i = \tan(\sqrt{x_2^i}), i = 1, \dots, r$, com ótimo global em $x_1^i = 0, i = 1, \dots, p$, $x_2^i = 0, i = 1, \dots, r$. O valor ótimo global da função objetivo do líder é $F(x, y) = 0$.

A Figura 5.3 mostra o comportamento da função objetivo do líder em relação às suas variáveis, assumindo resposta ótima do seguidor para todos os pontos desta instância e com parâmetros $p = 1$, $q = 1$ e $r = 1$ (como as componentes x_1 e x_2 tem apenas uma dimensão, seu índice é omitido no gráfico). É possível notar que,

considerando o seguidor sempre com resposta correta, a função objetivo do líder se reduz a um paraboloide centrado em $(0, 0)$.

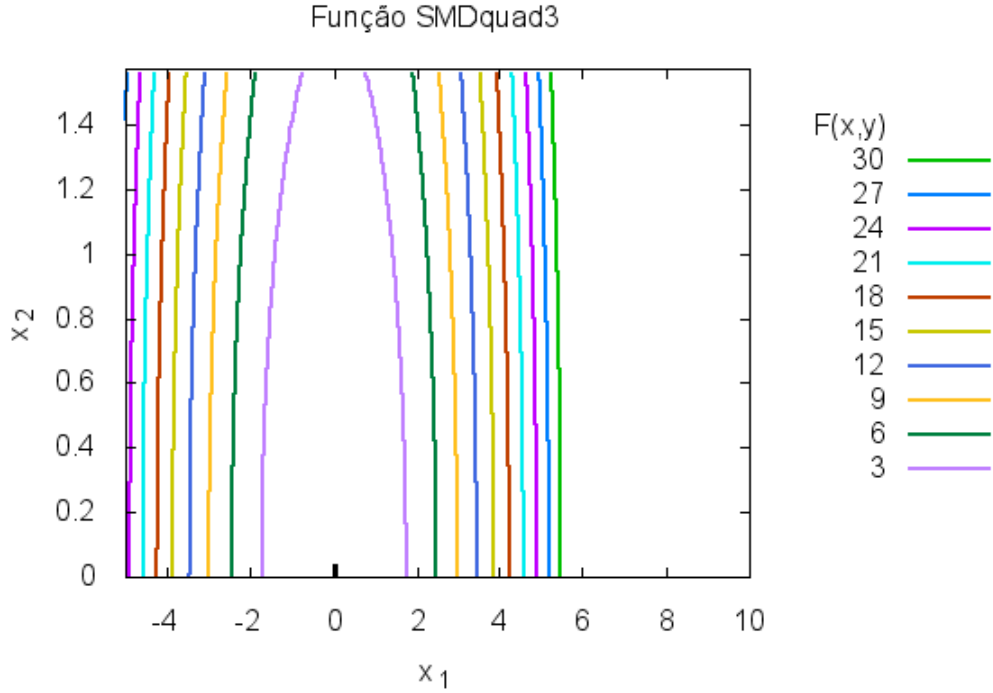


Figura 5.3: Função objetivo do Nível do Líder: SMDquad3

No caso de teste SMDquad4 a troca ocorre no termo $(|x_2^i| - \log(1 + y_2^i))$ que foi alterado para $(y_2^i - \log(x_2^i + 1))$, aproximando novamente do formato da função objetivo do líder e alterando apenas as restrições de “caixa”. Foi modificada ainda a função f_2 , com o termo $q + \sum_{i=1}^q ((y_1^i)^2 - \cos(2\pi y_1^i))$ mudado para $q + \sum_{i=1}^q ((y_1^i)^2 - 1)$.

SMD4	SMDquad4
$\min_x F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$ $F_1(x_1) = \sum_{i=1}^p (x_1^i)^2$ $F_2(y_1) = -\sum_{i=1}^q (y_1^i)^2$ $F_3(x_2, y_2) = \sum_{i=1}^r (x_2^i)^2 - \sum_{i=1}^r (x_2^i - \log(1 + y_2^i))^2$ s.t. $x_1 \in [-5, 10]$ $x_2 \in [-1, 1]$ $\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$ $f_1(x_1, x_2) = \sum_{i=1}^p (x_1^i)^2$ $f_2(y_1) = q + \sum_{i=1}^q ((y_1^i)^2 - \cos(2\pi y_1^i))$ $f_3(x_2, y_2) = \sum_{i=1}^r (x_2^i - \log(1 + y_2^i))^2$ s.t. $y_1 \in [-5, 10]$ $y_2 \in [0, e]$	$\min_x F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$ $F_1(x_1) = \sum_{i=1}^p (x_1^i)^2$ $F_2(y_1) = -\sum_{i=1}^q (y_1^i)^2$ $F_3(x_2, y_2) = \sum_{i=1}^r (x_2^i)^2 - \sum_{i=1}^r (y_2^i - \log(x_2^i + 1))^2$ s.t. $x_1 \in [-5, 10]$ $x_2 \in [0, e]$ $\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$ $f_1(x_1, x_2) = \sum_{i=1}^p (x_1^i)^2$ $f_2(y_1) = q + \sum_{i=1}^q ((y_1^i)^2 - 1)$ $f_3(x_2, y_2) = \sum_{i=1}^r (y_2^i - \log(x_2^i + 1))^2$ s.t. $y_1 \in [-5, 10]$ $y_2 \in [-1, 1]$

Tabela 5.5: SMD4 x SMDquad4

Agora a relação de otimalidade líder-seguidor é $y_1^i = 0$, $i = 1, \dots, q$, $y_2^i = \log(1 + x_2^i)$, $i = 1, \dots, r$, com ótimo global em $x_1^i = 0$, $i = 1, \dots, p$, $x_2^i = 0$, $i = 1, \dots, r$. O valor ótimo global da função objetivo do líder é $F(x, y) = 0$.

A Figura 5.4 mostra o comportamento da função objetivo do líder em relação às suas variáveis, assumindo resposta ótima do seguidor para todos os pontos desta instância e com parâmetros $p = 1$, $q = 1$ e $r = 1$ (como as componentes x_1 e x_2 tem apenas uma dimensão, seu índice é omitido no gráfico). É possível notar que, considerando o seguidor sempre com resposta correta, a função objetivo do líder se reduz a um paraboloide centrado em $(0, 0)$.

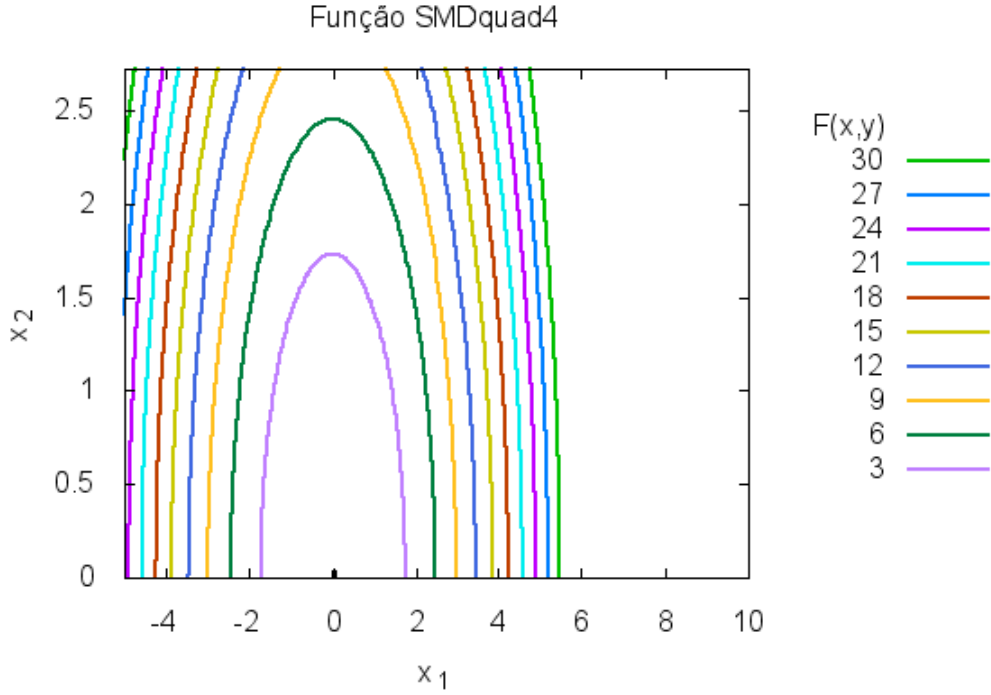


Figura 5.4: Função objetivo do Nível do Líder: SMDquad4

No caso de teste SMDquad5 a troca ocorre no termo $(|x_2^i| - (y_2^i)^2)$ que foi alterado para $(y_2^i - \sqrt{x_2^i})$, aproximando novamente do formato da função objetivo do líder e alterando apenas as restrições de “caixa”. Modificou-se ainda a função f_2 , com o termo $\sum_{i=1}^{q-1} ((y_1^{i+1} - (y_1^i)^2)^2 + (y_1^i - 1)^2)$ mudado para $\sum_{i=1}^{q-1} ((y_1^{i+1} - y_1^i)^2 + (y_1^i - 1)^2)$.

SMD5	SMDquad5
$\min_x F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$ $F_1(x_1) = \sum_{i=1}^p (x_1^i)^2$ $F_2(y_1) = -\sum_{i=1}^{q-1} ((y_1^{i+1} - (y_1^i)^2)^2 + (y_1^i - 1)^2)$ $F_3(x_2, y_2) = \sum_{i=1}^r (x_2^i)^2 - \sum_{i=1}^r (x_2^i - (y_2^i)^2)^2$ s.t. $x_1 \in [-5, 10]$ $x_2 \in [-5, 10]$ $\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$ $f_1(x_1, x_2) = \sum_{i=1}^p (x_1^i)^2$ $f_2(y_1) = \sum_{i=1}^{q-1} ((y_1^{i+1} - (y_1^i)^2)^2 + (y_1^i - 1)^2)$ $f_3(x_2, y_2) = \sum_{i=1}^r (x_2^i - (y_2^i)^2)^2$ s.t. $y_1 \in [-5, 10]$ $y_2 \in [-5, 10]$	$\min_x F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$ $F_1(x_1) = \sum_{i=1}^p (x_1^i)^2$ $F_2(y_1) = -\sum_{i=1}^{q-1} ((y_1^{i+1} - y_1^i)^2 + (y_1^i - 1)^2)$ $F_3(x_2, y_2) = \sum_{i=1}^r (x_2^i)^2 - \sum_{i=1}^r (y_2^i - \sqrt{x_2^i})^2$ s.t. $x_1 \in [-5, 10]$ $x_2 \in [0, 10]$ $\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$ $f_1(x_1, x_2) = \sum_{i=1}^p (x_1^i)^2$ $f_2(y_1) = \sum_{i=1}^{q-1} ((y_1^{i+1} - y_1^i)^2 + (y_1^i - 1)^2)$ $f_3(x_2, y_2) = \sum_{i=1}^r (y_2^i - \sqrt{x_2^i})^2$ s.t. $y_1 \in [-5, 10]$ $y_2 \in [-5, 10]$

Tabela 5.6: SMD5 x SMDquad5

Agora a relação de otimalidade líder-seguidor é $y_1^i = 1$, $i = 1, \dots, q$, $y_2^i = \sqrt{x_2^i}$, $i = 1, \dots, r$, com ótimo global em $x_1^i = 0$, $i = 1, \dots, p$, $x_2^i = 0$, $i = 1, \dots, r$. O valor ótimo global da função objetivo do líder é $F(x, y) = 0$.

A Figura 5.5 mostra o comportamento da função objetivo do líder em relação às suas variáveis, assumindo resposta ótima do seguidor para todos os pontos desta instância e com parâmetros $p = 1$, $q = 1$ e $r = 1$ (como as componentes x_1 e x_2 tem apenas uma dimensão, seu índice é omitido no gráfico). É possível notar que, considerando o seguidor sempre com resposta correta, a função objetivo do líder se reduz a um paraboloide centrado em $(0, 0)$.

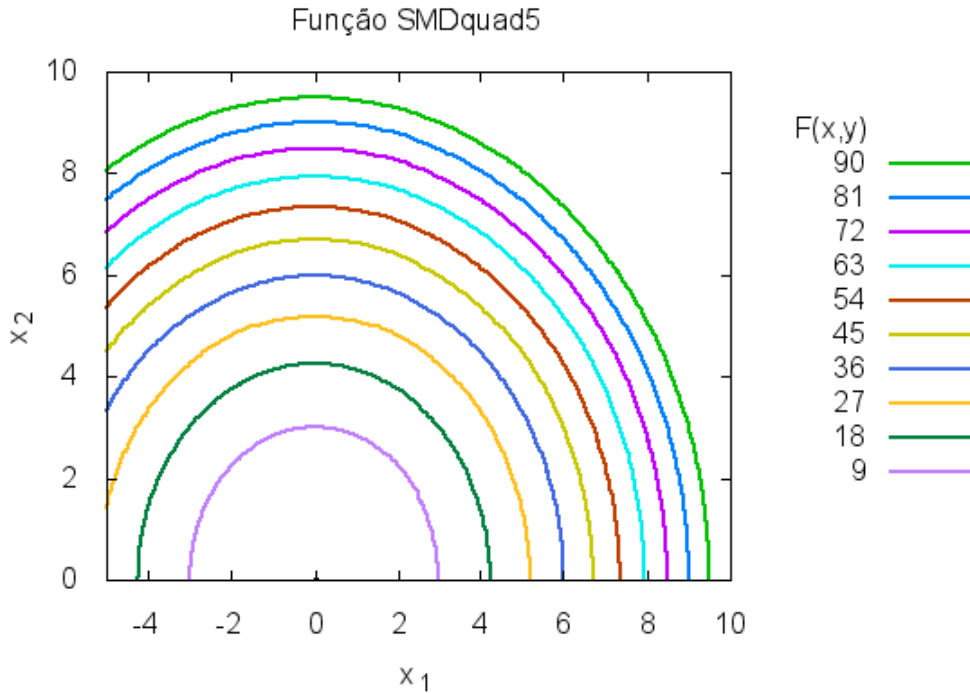


Figura 5.5: Função objetivo do Nível do Líder: SMDquad5

Assim como nas versões originais SMD, verifica-se que o desafio na resolução das instâncias SMDquad está no nível do seguidor (onde tem-se uma maior complexidade em relação ao formato da função) e nas implicações que uma resolução incorreta deste nível pode trazer ao problema líder. Como para a abordagem aqui proposta o nível do seguidor é resolvido exatamente, este fator é amenizado com as funções, não apresentando desafio tão grande à técnica.

5.2.2 Instâncias-Teste com Restrições Funcionais

Os casos de teste com restrições funcionais apresentados em Sinha et al. (2013) possuem complicações quando se considera uma conversão semelhante à que foi feita para as funções SMDquad. Estas complicações estão vinculadas à dificuldade em replicar o comportamento de restrições não-lineares, fazendo com que apresentem componente linear no seguidor e tenham, ainda assim, formato semelhante à função original. Desta forma, ao invés de adaptar as funções já existentes, optou-se pelo desenvolvimento de novas funções com restrições funcionais criadas de maneira semelhante ao que foi apresentado em Sinha et al. (2013).

Neste contexto, foram feitas alterações na metodologia com o objetivo de apresentar uma maior simplicidade na descrição das instâncias, abrindo-se mão das parcelas F_1 , F_2 , f_1 e f_2 . As parcelas F_1 e f_1 visam conferir dificuldade puramente à função objetivo do líder, não tendo qualquer aspecto vinculado à interação entre níveis (fator que caracteriza o BLPP), enquanto as parcelas F_2 e f_2 tem o mesmo propósito para a função objetivo do seguidor, não sendo de grande relevância dado que a resolução será exata. Ainda, os componentes x_1 e y_1 foram descartados, dado que não fazem parte das funções F_3 e f_3 e apenas o parâmetro r foi associado ao problema, com o número de dimensões de líder e seguidor possuindo valor r . As Tabelas 5.7, 5.8, 5.9 e 5.10 apresentam a formulação dos casos de teste Cquad1, Cquad2, Cquad3 e Cquad4, respectivamente.

A instância Cquad1 foi desenvolvida buscando gerar dificuldade em se atender às restrições do nível do líder, apresentando regiões viáveis no espaço de busca

isoladas por regiões inviáveis e sem haver um caminho contínuo entre as áreas viáveis. Há ainda um ponto de mínimo local na origem, o que dificulta a convergência.

Cquad1	
\min_x	$F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$
	$F_1(x_1) = 0$
	$F_2(y_1) = 0$
	$F_3(x_2, y_2) = \frac{1}{r}(\sum_{i=1}^r (y_2^i)^2 + 2 \sum_{i=1}^r (x_2^i)^3)$
s.t.	$x_2 \in [-1, 1]$
	$x_2^i \leq \sin(2y_2^i) + 1/4, \quad i \in \{1, \dots, r\}$
	$x_2^i \geq \sin(2y_2^i) - 1/4, \quad i \in \{1, \dots, r\}$
	$\min_y f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$
	$f_1(x_1, x_2) = 0$
	$f_2(y_1) = 0$
	$f_3(x_2, y_2) = \sum_{i=1}^r (y_2^i - x_2^i)^2$
s.t.	$y_2 \in [-1, 1]$

Tabela 5.7: Descrição da instância Cquad1.

Nesta elaboração, a relação de otimalidade líder-seguidor é $y_2^i = x_2^i$, $i = 1, \dots, r$, com ótimo global em $x_2^i = -1$, $i = 1, \dots, r$. O valor ótimo global da função objetivo do líder é $F(x, y) = -1$.

A Figura 5.6 mostra o comportamento da função objetivo do líder em relação às suas variáveis, assumindo resposta ótima do seguidor para todos os pontos desta instância e com parâmetro $r = 2$. É possível notar que, considerando o seguidor sempre com resposta correta, o problema no nível do líder apresenta regiões viáveis (em branco) e inviáveis (em cinza), havendo porções viáveis isoladas por porções inviáveis. O valor é crescente, a partir de valores positivos com um mínimo local em $(0, 0)$, região de crescimento em torno de $(-0.4, -0.4)$ e mínimo global em $(-1, -1)$.

A instância Cquad2 foi desenvolvida de forma a intensificar a interação entre líder e seguidor com a utilização de uma restrição capaz de controlar, indiretamente, o valor atribuído às variáveis do seguidor. Há também um ponto mínimo local na origem tornando a convergência mais complexa.

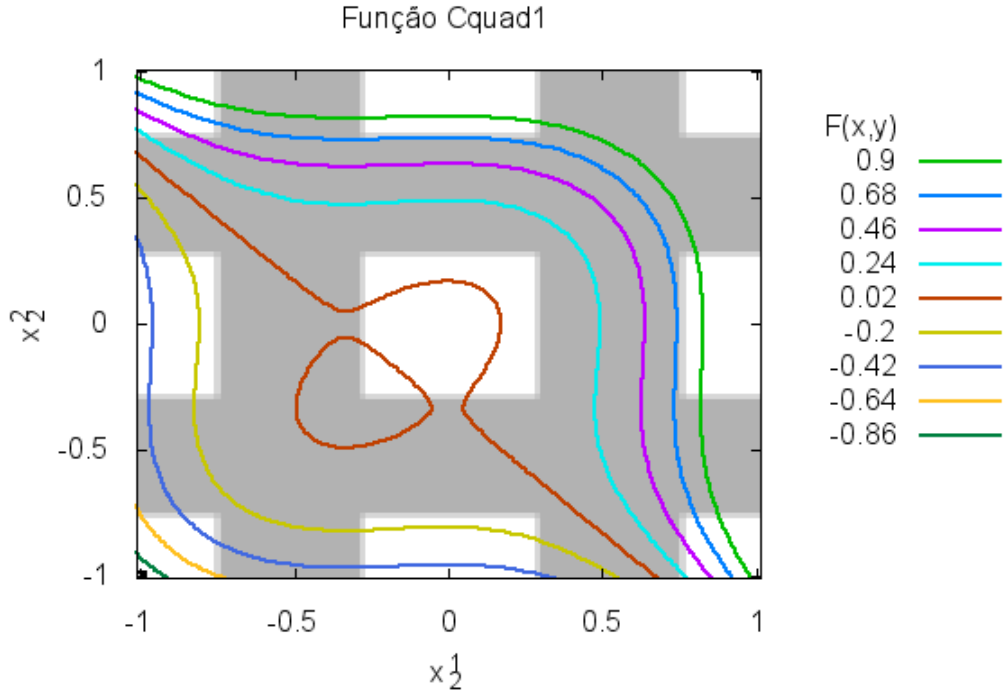


Figura 5.6: Função objetivo do Nível do Líder: Cquad1.

Cquad2	
\min_x	$F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$
	$F_1(x_1) = 0$
	$F_2(y_1) = 0$
	$F_3(x_2, y_2) = \frac{1}{r} (\sum_{i=1}^r x_2^i - (y_2^i)^2)$
s.t.	$x_2 \in [-5, 5]$
	$\sum_{i=1}^r (x_2^i)^2 \leq 4r$
\min_y	$f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$
	$f_1(x_1, x_2) = 0$
	$f_2(y_1) = 0$
	$f_3(x_2, y_2) = \frac{1}{r} \sum_{i=1}^r (y_2^i)^2$
s.t.	$y_2 \in [0, 5]$
	$\sum_{i=1}^r y_2^i \geq \sum_{i=1}^r x_2^i $

Tabela 5.8: Descrição da instância Cquad2.

Neste caso, a relação de otimalidade líder-seguidor é $y_2^i = \frac{1}{r} \sum_{j=1}^r x_2^j$, $i = 1, \dots, r$, com ótimo global em $x_2^i = 2$ ou $x_2^i = -2$, $i = 1, \dots, r$. O valor ótimo global da função objetivo do líder é $F(x, y) = -2$.

A Figura 5.7 mostra o comportamento da função objetivo do líder em relação às suas variáveis, assumindo resposta ótima do seguidor para todos os pontos desta instância e com parâmetro $r = 2$. É possível notar que, considerando o seguidor sempre com resposta correta, o problema no nível do líder apresenta regiões viáveis

(em branco) e inviáveis (em cinza), com a viabilidade do espaço de busca reduzida a uma circunferência de centro $(0,0)$ e raio $2\sqrt{2}$. O valor de função objetivo é inicialmente crescente a partir da origem, que é mínimo local, com decréscimo a partir da curva de nível que inclui o ponto $(1,1)$, tendo mínimos globais nos pontos $(2,2)$, $(-2,2)$, $(2,-2)$ e $(-2,-2)$.

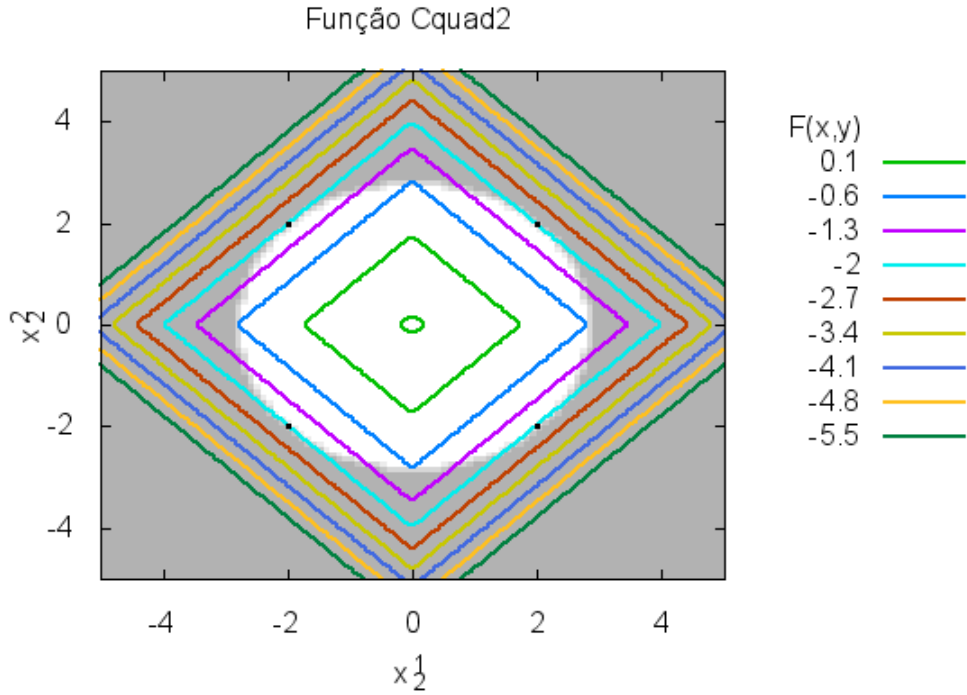


Figura 5.7: Função objetivo do Nível do Líder: Cquad2.

A instância Cquad3 foi desenvolvida especialmente para dificultar a análise completa do espaço de busca. Para isso, a função objetivo no nível do líder apresenta comportamento exponencial com o problema deste nível tendo ponto de mínimo local na origem e gradiente elevado em seu entorno. Há também uma relação de colaboração entre os níveis em relação às restrições, tornando mais complexo o comportamento da instância.

Nesta elaboração, a relação de otimalidade líder-seguidor é $y_2^1 = \frac{3|x_2^{(2)}|}{2}$ para a variável de índice 1, $y_2^i = \frac{(3|x_2^{i-1}| + 3|x_2^{i+1}|)}{4}$, $i \in \{2, \dots, r-1\}$ para variáveis intermediárias e $y_2^r = \frac{3|x_2^{r-1}|}{2}$ para a última variável, com ótimo global em $x_2^i = 4$, $i = 1, \dots, r$. O valor ótimo global da função objetivo do líder é $F(x, y) = -14$.

Cquad3	
\min_x	$F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$
	$F_1(x_1) = 0$
	$F_2(y_1) = 0$
	$F_3(x_2, y_2) = (-2^{1/r} \sum_{i=1}^r (y_2^i) + \frac{1}{r} \sum_{i=1}^r (2(x_2^i + 1)^2))$
s.t.	$x_2 \in [0, 10]$
	$(x_2^i)^2 \leq -\frac{1}{r} \sum_{j=1}^r (y_2^j) + 22$
\min_y	$f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$
	$f_1(x_1, x_2) = 0$
	$f_2(y_1) = 0$
	$f_3(x_2, y_2) = \frac{1}{r} \sum_{i=1}^r (y_2^i)^2$
s.t.	$y_2 \in [0, 10]$
	$y_2^1 \geq \frac{3 x_2^2 }{2}$
	$y_2^r \geq \frac{3 x_2^{r-1} }{2}$
	$y_2^i \geq \frac{(3 x_2^{i-1} + 3 x_2^{i+1})}{4}, i \in \{2, \dots, r-1\}$

Tabela 5.9: Descrição da instância Cquad3.

A Figura 5.8 mostra o comportamento da função objetivo do líder em relação às suas variáveis, assumindo resposta ótima do seguidor para todos os pontos desta instância e com parâmetro $r = 2$. É possível notar que, considerando o seguidor sempre com resposta correta, o problema à nível do líder apresenta regiões viáveis (em branco) e inviáveis (em cinza), com a viabilidade do espaço de busca reduzida ao quadrante contendo $(0, 0)$. O valor é inicialmente crescente a partir da origem, que é mínimo local, com decréscimo em torno do ponto $(2, 2)$, tendo o mínimo global em $(4, 4)$.

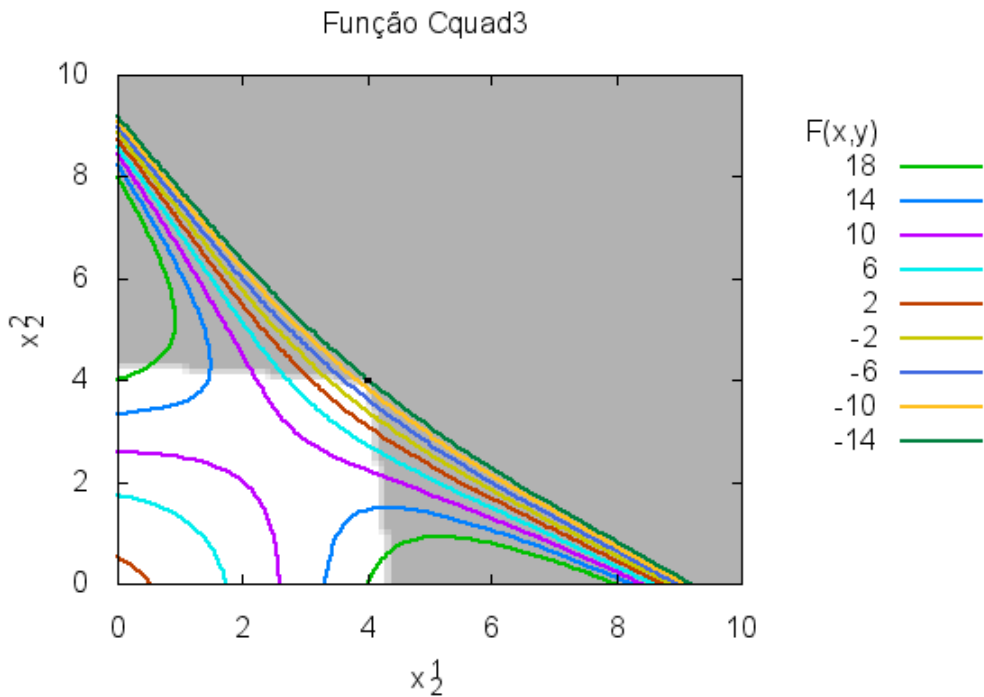


Figura 5.8: Função objetivo do Nível do Líder: Cquad3.

A instância Cquad4 foi desenvolvida tendo em vista as restrições no seguidor. Estas, apesar de não impedirem a viabilidade das soluções, tornam o comportamento do seguidor pouco previsível, o que reflete na função objetivo do líder de diversos máximos e mínimos locais.

Cquad4	
\min_x	$F(x, y) = F_1(x_1) + F_2(y_1) + F_3(x_2, y_2)$
	$F_1(x_1) = 0$
	$F_2(y_1) = 0$
	$F_3(x_2, y_2) = \frac{1}{r} \sum_{i=1}^r (x_2^i + 10 y_2^i)$
s.t.	$x_2 \in [-2, 2]$
\min_y	$f(x, y) = f_1(x_1, x_2) + f_2(y_1) + f_3(x_2, y_2)$
	$f_1(x_1, x_2) = 0$
	$f_2(y_1) = 0$
	$f_3(x_2, y_2) = \sum_{i=1}^r (y_2^i)^2 - 10y_2^i$
s.t.	$y_2 \in [-1, 1]$
	$y_2^i \leq \sin(1/2(x_2^i - 1)\pi), \quad i \in \{1, \dots, r\}$
	$y_2^i \leq \sin(2(x_2^i + 1/2)\pi), \quad i \in \{1, \dots, r\}$

Tabela 5.10: Descrição da instância Cquad4.

A relação de otimalidade líder-seguidor é $y_2^i = \sin(1/2(x_2^i - 1)\pi)$ quando $\sin(1/2(x_2^i - 1)\pi) \leq \sin(2(x_2^i + 1/2)\pi)$ ou $y_2^i = \sin(2(x_2^i + 1/2)\pi)$ em caso contrário, $i = 1, \dots, r$, com ótimo global em $x_2^i = 1$ ou $x_2^i = -1$, $i = 1, \dots, r$. O valor ótimo global da função objetivo do líder é $F(x, y) = 1$.

A Figura 5.9 mostra o comportamento da função objetivo do líder em relação às suas variáveis, assumindo resposta ótima do seguidor para todos os pontos desta instância e com parâmetro $r = 2$. É possível notar que, considerando o seguidor sempre com resposta correta, o problema à nível do líder é viável para todos os pontos do espaço de busca. O valor da função é variado por toda a região, com mínimos e máximos locais espalhados, apresentando mínimos globais em $(1, 1)$, $(-1, 1)$, $(1, -1)$ e $(-1, -1)$.

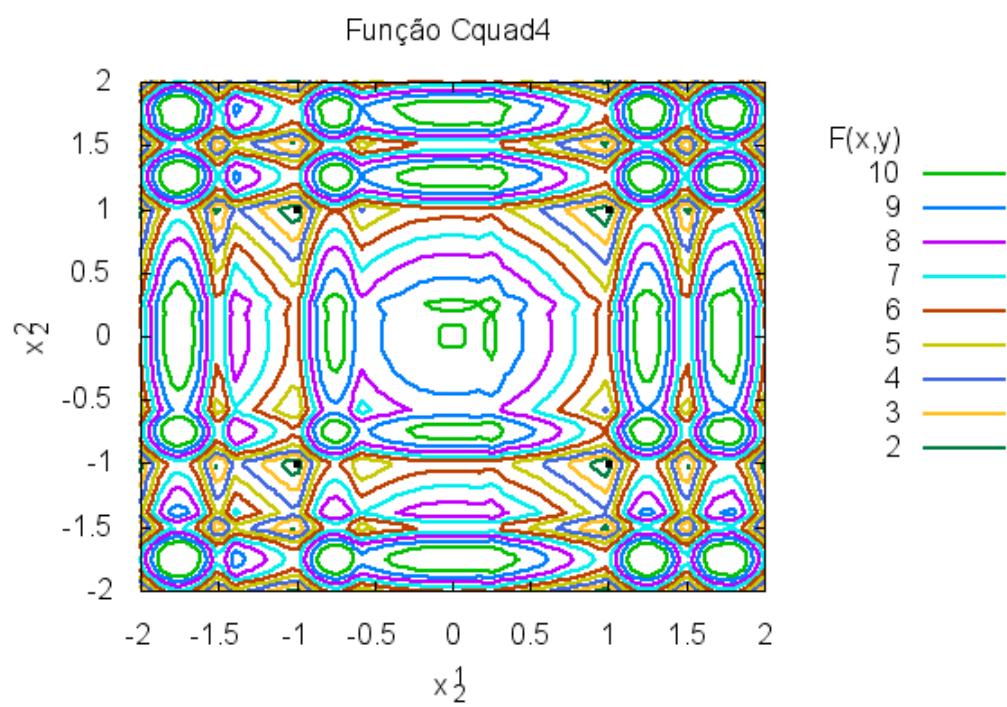


Figura 5.9: Função objetivo do Nível do Líder: Cquad4.

Capítulo 6

Experimentos Computacionais

Neste capítulo é apresentada uma série de experimentos com o intuito tanto de validar a utilização do algoritmo de Lemke na resolução do problema no nível do seguidor, quanto de mostrar o desempenho da abordagem proposta DE+LEMKE na resolução do problema de dois níveis.

Para isso são utilizados os casos de teste presentes no Capítulo 5, com os casos escaláveis definidos em 10, 20 e 30 dimensões. Neste sentido, para os casos de teste SMDquad1 a SMDquad5 são utilizadas as configurações de parâmetros $(p = 3, q = 3, r = 2)$, $(p = 6, q = 6, r = 4)$ e $(p = 9, q = 9, r = 6)$, no caso de teste SMD6 tem-se $(p = 3, q = 1, r = 2, s = 2)$, $(p = 6, q = 2, r = 4, s = 4)$ e $(p = 9, q = 3, r = 6, s = 6)$ e nos casos com restrição funcional Cquad1 a Cquad4 os parâmetros variam entre $(r = 5)$, $(r = 10)$ e $(r = 15)$.

Ainda, visando facilitar a visualização em gráficos e tabelas, é apresentada apenas a dimensão de cada problema escalável entre parenteses ao invés da configuração utilizada (vista no parágrafo anterior), o que evita um excesso de informações ao se identificar as instâncias.

Seguem os parâmetros utilizados e os três conjuntos de testes para os quais são apresentados o custo médio do algoritmo de Lemke em operações de pivota-mento, uma comparação entre a utilização do algoritmo de Lemke e uma metaheurística DE no nível do seguidor e, por fim, uma comparação de desempenho da técnica DE+LEMKE com uma variação desta denominada R-DE+LEMKE (defi-

nida adiante no texto) e a opção por se resolver o seguidor via evolução diferencial, DE+DE.

6.1 Ajuste de parâmetros

A definição dos parâmetros foi feita via ferramenta irace (López-Ibáñez et al., 2011). Esta ferramenta visa determinar, através de conjuntos independentes de experimentos, qual a melhor combinação de valores a serem atribuídos aos diferentes parâmetros existentes em uma dada técnica.

Para isso, o irace utiliza o conceito de “corridas” que são séries de experimentos realizados sobre amostras com intuito de avaliação das mesmas. São realizadas sucessivas “corridas” permitindo avaliar possíveis configurações de parâmetros e determinar novas combinações através do desempenho das já avaliadas.

Para utilização da ferramenta, foi definido um subconjunto de casos de teste entre os que foram apresentados no Capítulo 5 e selecionado o teste estatístico teste-F. Com estas configurações e através de 15.000 experimentos foram determinados os seguintes valores para cada um dos parâmetros (seguidos pelos intervalos considerados no teste):

- Tamanho da população (NP): $20 \sim [15,80]$
- Fator de ponderação para mutação (F): $0.7 \sim [0.5,1.0]$
- Fator de cruzamento (CR): $0.6 \sim [0.3,1]$

Estes parâmetros selecionados foram utilizados em todos os experimentos realizados com a técnica DE+LEMKE, bem como nas demais utilizações da DE aplicadas ao problema no nível do líder.

6.2 Análise de custo computacional do algoritmo de Lemke

Para a análise do custo computacional produzido pela escolha do algoritmo de Lemke foram coletadas informações sobre número de operações de pivotamento em 50 execuções da técnica DE+Lemke em cada caso de teste proposto, considerando

como critérios de parada 6.000 avaliações de função do líder ou 18.000.000 operações de pivotamento.

A Tabela 6.1 apresenta a dimensão para líder e seguidor em cada um dos casos de teste do grupo obtido da literatura, bem como o número de restrições funcionais destes (entre parênteses o número de restrições total, incluindo restrições de “caixa”) e números máximo, médio e mínimo de operações de pivotamento para cada execução do algoritmo de Lemke com o desvio-padrão desta medida.

Tabela 6.1: Operações de pivotamento e característica dos problemas - Instâncias da literatura

Teste	Líder		Seguidor		Matriz Lemke	Operações de Pivotamento			
	Dim	Restr	Dim	Restr		Max	Média	Min	Desvio- Padrão
Pr1	2	3(3)	2	0(4)	6x14	4	3.00	1	0.08
Pr2	2	0(0)	2	3(7)	9x20	7	4.22	1	0.42
Pr3	2	1(1)	2	2(10)	14x30	6	3.05	2	0.48
Pr4	1	0(0)	1	3(5)	6x14	3	2.07	1	0.28
Pr5	2	1(1)	2	2(6)	8x18	7	3.04	3	0.31
Pr6	1	0(0)	2	4(7)	9x20	5	4.07	3	0.28
Pr7	1	0(0)	1	0(2)	3x8	1	1.00	1	0.00
Pr8	1	0(0)	1	3(6)	7x16	3	2.00	1	0.09
Pr9	2	0(2)	3	3(6)	9x20	5	4.11	2	0.35
Pr10	2	1(3)	3	3(6)	9x20	5	4.19	2	0.41
Pr11	4	6(10)	2	4(6)	8x18	7	2.33	1	2.33
Pr12	10	2(22)	6	7(19)	25x52	38	19.93	7	2.27
Pr13	4	1(9)	4	4(12)	16x34	12	8.69	4	1.10
Pr14	2	1(5)	2	2(6)	10x22	8	3.56	2	0.70
Pr15	1	0(1)	2	4(6)	8x18	6	4.21	2	0.63
Pr16	2	0(0)	2	2(4)	6x14	6	1.29	1	0.48
Pr17	1	0(0)	2	2(5)	7x16	4	3.99	1	0.12
Pr18	1	0(0)	2	2(5)	7x16	4	3.11	1	0.32

Pode-se notar que, com exceção do problema Pr12, não há um padrão bem determinado para a dimensão do problema, número de restrições ou mesmo quando se verifica o funcional aplicado, sendo muito provavelmente o número de operações mais ligado à estrutura da função do que a estes fatores. Apesar disto, deve-se levar em conta a pouca variação entre número de restrições e dimensão dos problemas analisados, sendo possível, neste sentido, observar algumas tendências. Estas podem ser melhor observadas nos gráficos contidos nas Figuras 6.1 e 6.2.

Com relação ao problema Pr12 pode-se verificar que a dimensão e o número de restrições elevado (comparado aos demais problemas) refletiu, aparentemente,

em um número muito maior de operações de pivotamento.

O gráfico da Figura 6.1 apresenta as operações de pivotamento em barras, com valor máximo, médio e mínimo para cada problema. Apresenta também a variação da dimensão e número de restrições funcionais nestes mesmos problemas considerando o nível do líder. De maneira análoga, as mesmas informações de pivotamento são apresentadas no gráfico da Figura 6.2, mas com dimensão e restrições referentes ao nível do seguidor destes problemas.

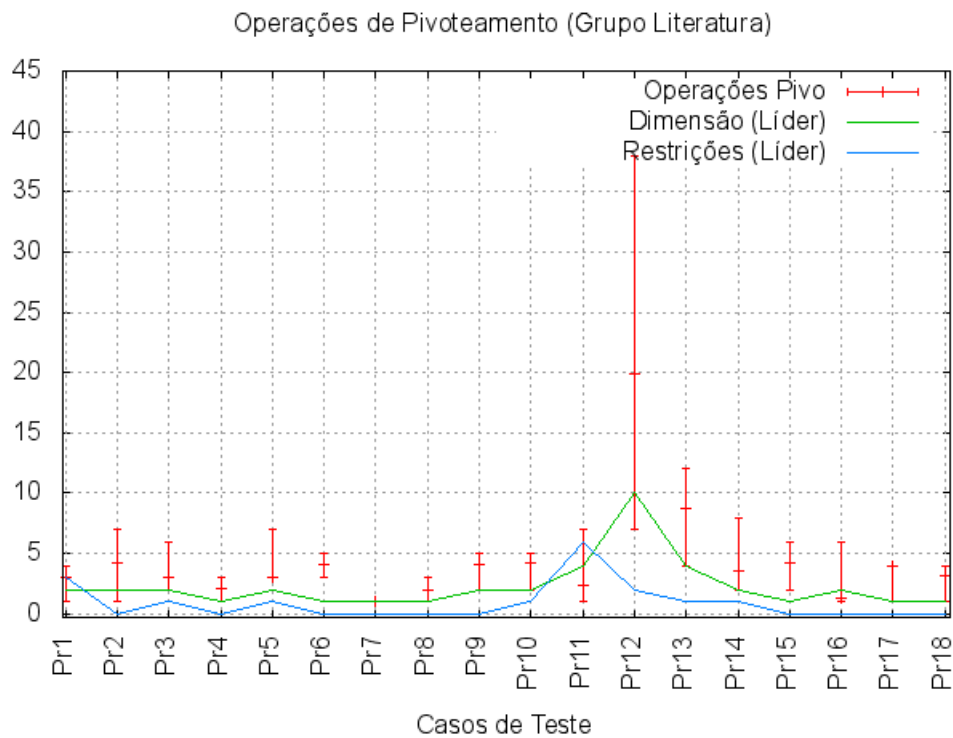


Figura 6.1: Literatura - Operações de pivotamento x Dimensão x Restrições (Líder)

Observando os gráficos das Figuras 6.1 e 6.2, a dimensão do problema aparenta ter certo papel na variação do número de operações, assim como o número de restrições funcionais presentes no seguidor. O número de restrições funcionais do líder, por outro lado, parece ser fator pouco importante, especialmente dado que seu maior valor ocorre no Pr11, onde não há um número de operações acima da média geral, enquanto nas instâncias de maior quantidade de operações (Pr12 e Pr13) o número de restrições funcionais está dentro da média dos outros proble-

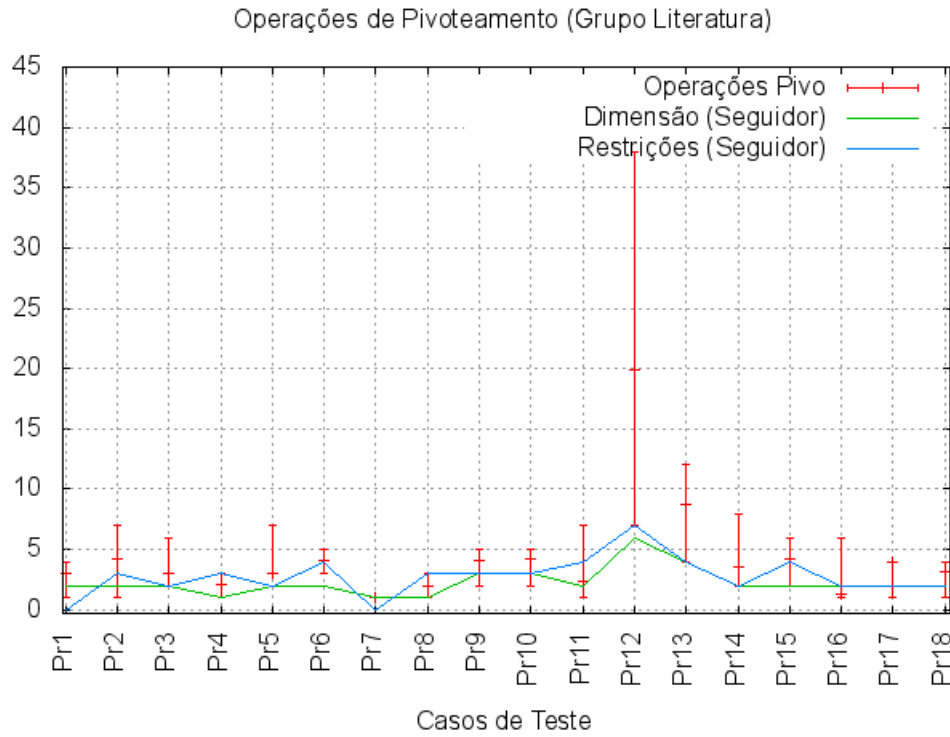


Figura 6.2: Literatura - Operações de pivoteamento x Dimensão x Restrições (Seguidor)

mas. É possível notar que, apesar do comportamento ao incluir as restrições de “caixa” às restrições do líder ser mais próximo do número de operações de pivoteamento realizadas, estas restrições praticamente não tem influência na construção do problema do seguidor e são diretamente influenciadas pela dimensão do líder, sendo esta uma explicação mais plausível.

A pouca influência das restrições funcionais do líder é, de certa forma, esperada, dado que a dimensão da matriz utilizada no algoritmo de Lemke é afetada apenas pelos elementos do seguidor, podendo, no máximo, o líder influir de maneira indireta na convergência do algoritmo através dos valores atribuídos à matriz. A relevância da dimensão do líder, da mesma forma, é bem questionável; mas como esta vem atrelada, na maior parte dos problemas deste conjunto, à variação da dimensão do próprio seguidor, torna-se difícil verificar este fator com clareza.

Análogo ao que foi feito anteriormente, a Tabela 6.2 apresenta a dimensão do problema para o nível do líder e do seguidor em cada um dos casos de teste

do grupo de instâncias escaláveis. Apresenta ainda o número de restrições (sem e com restrições de “caixa”) e números máximo, médio e mínimo de operações de pivotamento para cada execução do algoritmo de Lemke acompanhado do desvio-padrão desta medida.

Tabela 6.2: Operações de pivotamento e característica dos problemas - Instâncias escaláveis

Teste	Líder		Seguidor		Matriz Lemke	Operações de Pivotamento			
	Dim	Restr	Dim	Restr		Max	Média	Min	Desvio- Padrão
SMDquad1(10)	5	0(10)	5	0(10)	20x42	4	1.99	1	0.20
(20)	10	0(20)	10	0(20)	40x82	8	3.95	1	0.36
(30)	15	0(30)	15	0(30)	60x122	10	5.77	1	0.93
SMDquad2(10)	5	0(10)	5	0(10)	20x42	4	1.97	1	0.18
(20)	10	0(20)	10	0(20)	40x82	7	3.85	1	0.44
(30)	15	0(30)	15	0(30)	60x122	10	5.79	1	0.60
SMDquad3(10)	5	0(10)	5	0(10)	20x42	4	1.57	1	0.50
(20)	10	0(20)	10	0(20)	40x82	6	3.38	1	0.76
(30)	15	0(30)	15	0(30)	60x122	7	5.46	1	0.75
SMDquad4(10)	5	0(10)	5	0(10)	20x42	4	2.01	1	0.27
(20)	10	0(20)	10	0(20)	40x82	8	3.96	1	0.60
(30)	15	0(30)	15	0(30)	60x122	12	6.00	1	0.85
SMDquad5(10)	5	0(10)	5	0(10)	20x42	5	4.43	3	0.65
(20)	10	0(20)	10	0(20)	40x82	10	9.39	6	0.77
(30)	15	0(30)	15	0(30)	60x122	15	14.47	9	0.75
SMD6(10)	5	0(10)	5	0(10)	20x42	4	1.97	1	0.21
(20)	10	0(20)	10	0(20)	40x82	7	3.90	1	0.41
(30)	15	0(30)	15	0(30)	60x122	10	5.85	1	0.57
Cquad1(10)	5	10(20)	5	0(10)	20x42	11	6.79	2	1.78
(20)	10	20(40)	10	0(20)	40x82	18	10.78	3	2.77
(30)	15	30(60)	15	0(30)	60x122	24	15.32	4	3.75
Cquad2(10)	5	1(11)	5	1(11)	16x34	11	6.00	6	0.01
(20)	10	1(21)	10	1(21)	31x64	11	11.00	11	0.00
(30)	15	1(31)	15	1(31)	46x94	16	16.00	16	0.00
Cquad3(10)	5	5(15)	5	5(15)	20x42	11	10.00	4	0.12
(20)	10	10(30)	10	10(30)	40x82	21	19.99	9	0.21
(30)	15	15(45)	15	15(45)	60x122	31	29.98	9	0.31
Cquad4(10)	5	0(10)	5	10(20)	30x62	20	13.38	10	3.57
(20)	10	0(20)	10	20(40)	60x122	40	31.43	20	5.39
(30)	15	0(30)	15	30(60)	90x182	60	50.43	30	4.24

Neste experimento o aumento da dimensão em uma mesma instância levou em todos os casos a um incremento no número de operações de pivotamento, tanto nos casos sem restrições funcionais quanto para as instâncias em que estas estão presentes. Da mesma forma, casos com maior número de restrições funcionais no seguidor também levam a um maior número destas operações.

O gráfico da Figura 6.3 apresenta as operações de pivotamento em barras,

com valor máximo, médio e mínimo para os casos de teste escaláveis apenas com restrições de “caixa”. Apresenta também a variação da dimensão para os problemas do líder e do seguidor (mesmo valor nestes experimentos). De maneira análoga, as mesmas informações de pivotamento são apresentadas no gráfico da Figura 6.4, com dimensão de líder e seguidor e as restrições funcionais de cada nível individualizadas, sendo desta vez para os casos de teste escaláveis com restrições funcionais.

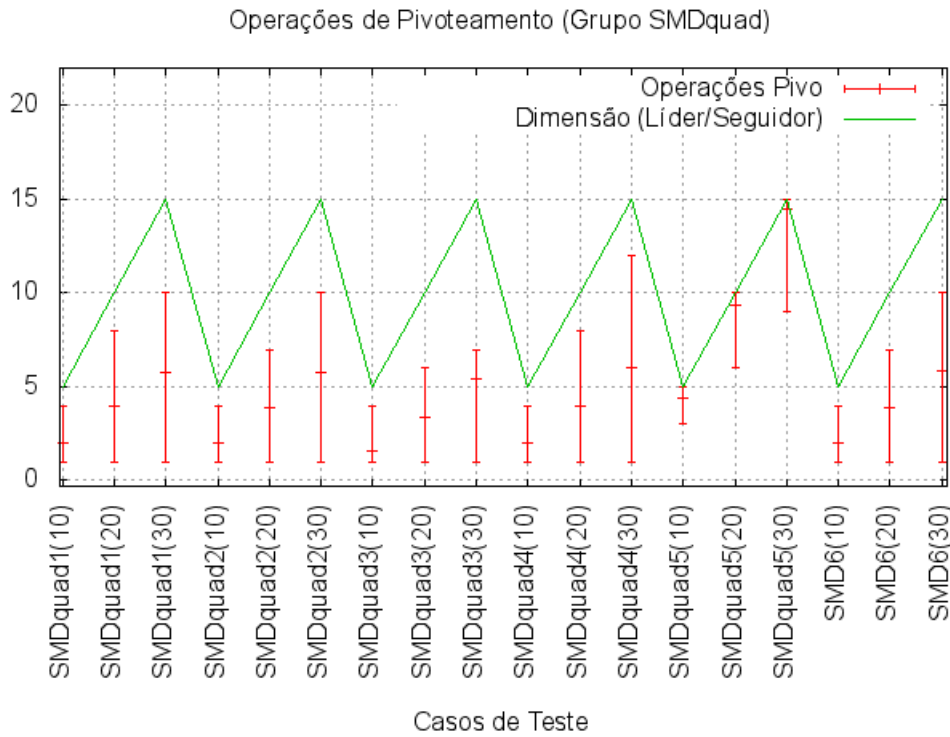


Figura 6.3: SMDquad - Operações de pivotamento x Dimensão (Líder/Seguidor)

Considerando os casos de teste com restrições de “caixa” fica evidente o crescimento no número de operações de pivotamento ao se aumentar a dimensão. É possível notar também que há certo papel na estrutura da função, havendo em determinados casos, especialmente o SMDquad5, um maior número de operações não tendo este comportamento relação com número de restrições ou dimensão do problema.

Da mesma forma que para as instâncias apenas com restrições de “caixa”, o acréscimo da dimensão leva a um notável aumento no número de operações de

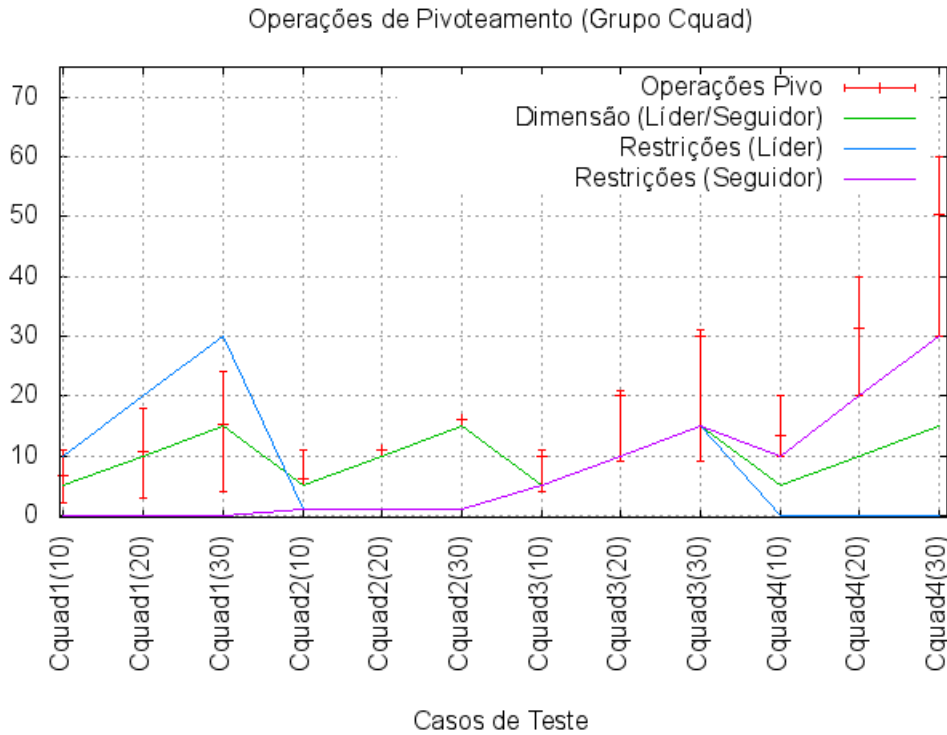


Figura 6.4: Cquad - Operações de pivoteamento x Dimensão (Líder/Seguidor) x Restrições (Líder/Seguidor)

pivotamento nas instâncias com restrições funcionais. Além deste, é pertinente notar o comportamento para as restrições do líder e do seguidor. Há um acréscimo no número médio de operações de pivotamento à medida que se aumenta o número de restrições funcionais para o seguidor, comportamento que não se observa para as restrições funcionais do líder. Até há um maior desvio em relação à média na instância Cquad1 (o que pode inclusive ser induzido pelo número maior de restrições funcionais no líder), mas não resultando em um acréscimo no valor médio.

Os casos de teste escaláveis se mostram, da mesma forma que nos casos de teste da literatura, com forte acoplamento entre dimensão do problema do líder e do seguidor, inviabilizando a análise do efeito da dimensão do líder nas operações de pivotamento. E, apesar de nos casos de teste com restrições de “caixa” ser possível aumentar a dimensão do problema do líder independentemente da do seguidor, este aumento não afeta a construção da matriz do método de Lemke, o que permitiria apenas uma análise enviesada.

Por fim, os gráficos das Figuras 6.5, 6.6 e 6.7 apresentam informações diretamente sobre a dimensão da matriz utilizada no método de Lemke, comparando-a ao número de operações de pivotamento máximo, médio e mínimo representado em barras. O tamanho da matriz é afetado pela dimensão do problema do seguidor, número de restrições no seguidor (incluindo aqui restrições de “caixa”), bem como número de variáveis que permitam valores negativos e número de restrições de igualdade (componente inexistente nos casos de teste utilizados).

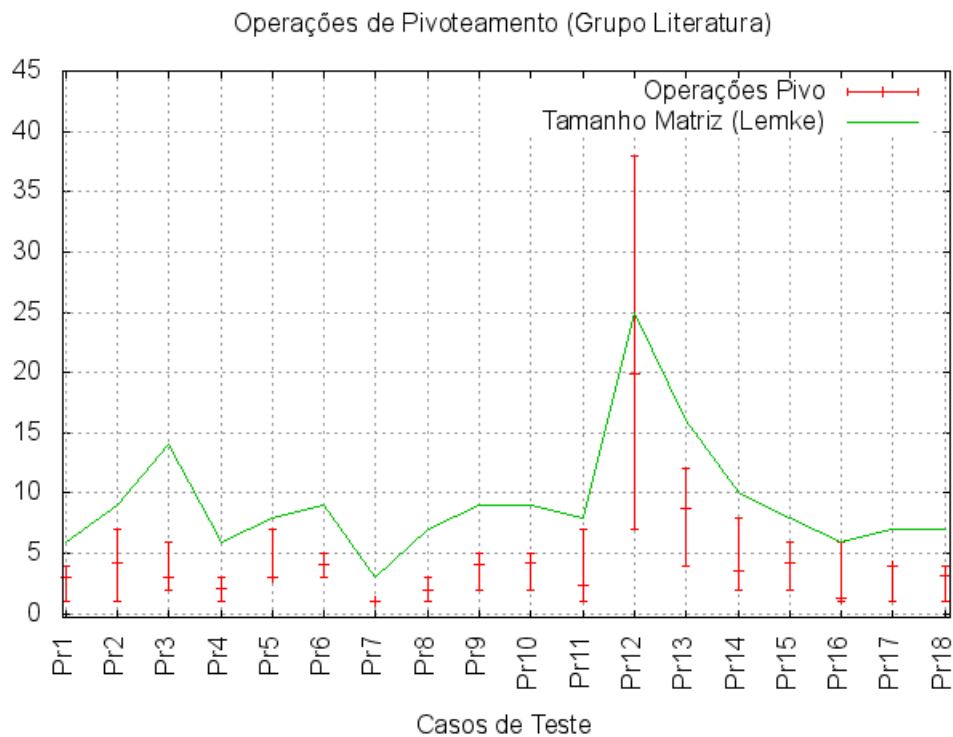


Figura 6.5: Literatura - Operações de pivotamento x Dimensão matriz de Lemke

Em relação aos casos teste da literatura (Figura 6.5) é pertinente notar que, apesar da baixa variação em número de dimensões e restrições, ao se considerar o tamanho da matriz utilizada no método de Lemke há uma maior correspondência com as operações de pivotamento quando comparadas aos fatores que influem neste tamanho individualmente. Por outro lado, pode-se observar que nem sempre esta correspondência é exata (como no caso Pr3 de segundo maior tamanho de matriz mas com apenas o 11º maior número de operações de pivotamento médio), tendo

ainda influência a forma da função em questão.

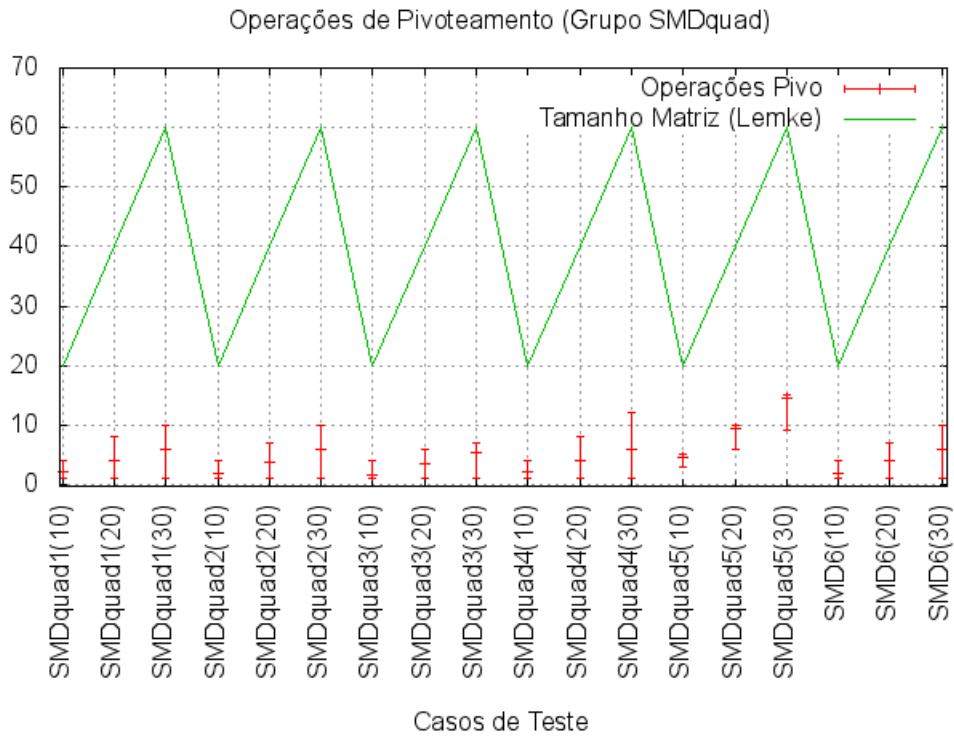


Figura 6.6: SMDquad - Operações de pivoteamento x Dimensão matriz de Lemke

Nos testes escaláveis apenas com restrições de “caixa” (Figura 6.6) não há grande novidade dado que o tamanho da matriz é influenciado exclusivamente pela dimensão do nível do seguidor (incluindo restrições de “caixa” e valores negativos). Como observado no gráfico da Figura 6.3 há uma correspondência direta no aumento da dimensão e crescimento do número de operações de pivoteamento, havendo variações entre casos de teste devido à forma das funções correspondentes.

Os testes escaláveis com restrições funcionais (Figura 6.7) são, das classes observadas, os que mostram maior correspondência em termos de tamanho de matriz e número de operações de pivoteamento. Os casos Cquad4, de maior dimensão da matriz, apresentaram número mais elevado de operações de pivoteamento, enquanto os casos Cquad1 e Cquad3 estão num patamar intermediário em ambas as medidas e os casos Cquad2 apresentam o menor destes valores.

Dadas estas observações é possível especular que, apesar dos atributos que influenciam individualmente a dimensão da matriz do método de Lemke terem

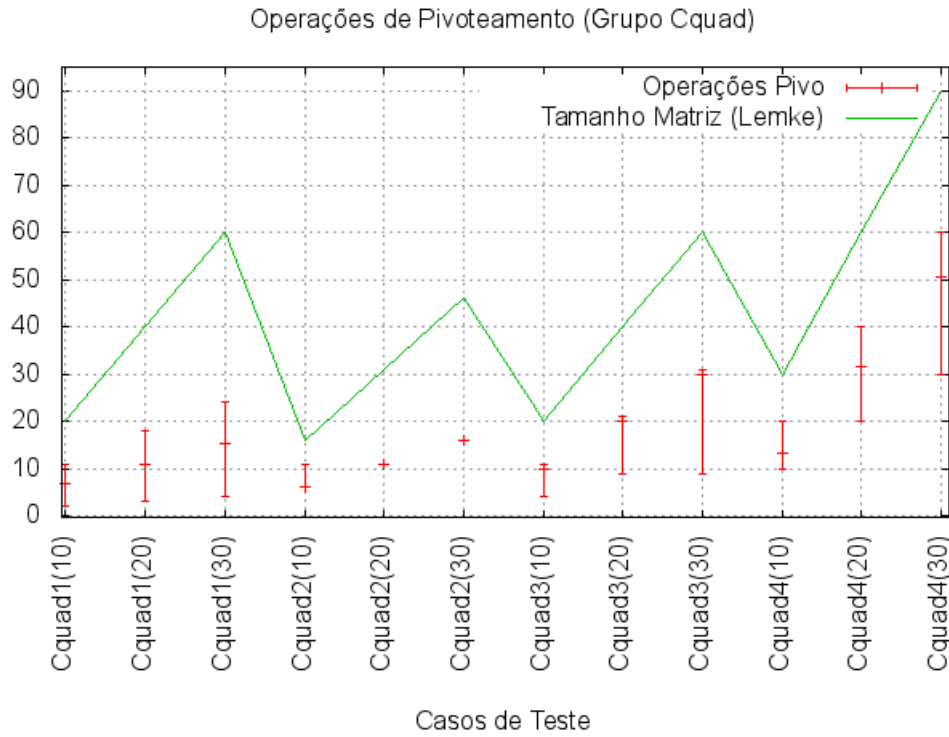


Figura 6.7: Cquad - Operações de pivotamento x Dimensão matriz de Lemke

papel no número de operações de pivotamento, não há uma única variável que se destaque, tendo os atributos que influem nesta medida pesos semelhantes. Por outro lado, a variação na dimensão do problema do seguidor costuma apresentar um maior destaque justamente por indiretamente induzir crescimento de outras variáveis, especialmente restrições de “caixa”, dado que a implementação explícita destas é necessária para a execução do método de Lemke. Ainda, há papel relacionado à forma da função na quantidade de operações realizadas, mas não a ponto de descaracterizar a tendência principal relacionada à dimensão da matriz.

6.3 Comparação da técnica DE+LEMKE a um algoritmo DE+DE

Para verificar a funcionalidade e possíveis vantagens e desvantagens de se optar por uma técnica de resolução exata, como o método de Lemke, em relação a uma metaheurística no nível do seguidor foi realizada a comparação do algoritmo DE+LEMKE com uma implementação DE+DE, sendo o líder resolvido através do

mesmo DE proposto por este trabalho e o seguidor através de uma implementação DE semelhante à utilizada em Angelo et al. (2013), incluindo parâmetros adotados.

Nas análise dos algoritmos optou-se por não comparar desempenho relacionado ao tempo de execução na utilização do método de Lemke com a implementação do DE. Esta escolha se deve à possível existência de implementações do DE mais eficientes para a resolução do nível do seguidor do que a aqui utilizada, bem como a existência de implementações do método de Lemke de menor custo computacional e maior robustez.

Desta forma, visando contabilizar de alguma maneira e analisar o custo de utilização de cada uma destas alternativas, optou-se por verificar o número de soluções geradas na implementação da DE, comparando-o às operações de pivotamento do método de Lemke. De fato, as métricas não são equivalentes, mas podem apresentar um indicativo da opção mais competitiva.

6.3.1 Instâncias da literatura

Como primeiro experimento foi verificada a capacidade de um algoritmo DE para o problema no nível do seguidor atingir as mesmas respostas fornecidas pelo método de Lemke. Para isso, foram realizadas 50 execuções do algoritmo DE+LEMKE em todas as instâncias, tendo, para cada solução encontrada na execução do método de Lemke (sem ocorrência de parada em raio), a aplicação da DE para o seguidor nas mesmas configurações do líder visando encontrar esta solução ótima ou outra que o valha (no caso de múltiplas soluções no seguidor).

Como critérios de parada foram utilizados para o nível do líder a exigência de proximidade relativa ao valor da melhor solução conhecida na literatura como 10^{-2} (precisão considerada suficiente para estes experimentos) ou 10.000 avaliações da função do líder. Para o problema do seguidor resolvido via DE, foi exigida uma proximidade relativa do valor de função objetivo, obtido via Lemke, de pelo menos 10^{-2} ou 1.000 iterações do DE.

Os resultados deste experimento utilizando os casos de teste do conjunto

obtido da literatura podem ser vistos na Tabela 6.3. Nela é possível observar o número médio de operações de pivotamento, excluindo aqui as execuções finalizadas com parada em raio (coluna OP), número médio de soluções geradas (coluna Soluções), número médio de chamadas realizados ao DE no nível do seguidor (coluna Cham.), porcentagem média das execuções do DE do seguidor onde não foi atingida a proximidade mínima da solução encontrada via método de Lemke (coluna (%) S/Sucesso), as médias do número de iterações médio necessário para convergência da DE para o seguidor por chamada realizada, desconsiderando os casos de insucesso (coluna Iter./Cham.) e a razão da média do número de soluções criadas pelo DE para o seguidor, comparando-o à média do número de operações de pivotamento do método de Lemke, considerando as mesmas configurações fornecidas pelo líder (coluna Sol./OP).

Tabela 6.3: Comparação DE x Lemke para instâncias da literatura

Teste	Lemke	DE				Sol./ O.P.
	O.P.	Soluções	Cham.	(%) S/ sucesso	Iter./ Cham.	
Pr1	461.12	15458.40	148.62	0.00	3.47	33.52
Pr2	769.80	3510664.20	187.86	5.93	563.56	4560.49
Pr3	3686.04	858306.00	1576.72	0.00	18.13	232.85
Pr4	2101.14	481281.60	1064.28	0.02	14.90	229.06
Pr5	220.06	39693.00	42.62	0.09	30.10	180.37
Pr6	232.18	18468.00	64.54	0.00	9.54	79.54
Pr7	54.98	15088213.80	95.70	52.54	4729.47	274430.95
Pr8	45.84	3841.20	28.12	0.00	4.55	83.80
Pr9	668.20	2307522.00	198.68	3.66	350.46	3453.34
Pr10	36882.28	2584965332.40	9231.48	93.31	8399.84	70086.92
Pr11	1345.28	215337.00	611.12	0.00	11.75	160.07
Pr12	20864.36	6240091.20	1095.64	1.31	176.76	299.08
Pr13	1757.14	796181.40	249.74	0.61	100.18	453.11
Pr14	5326.42	1401424.20	2350.56	0.00	19.87	263.11
Pr15	233.60	5856.60	68.92	0.00	2.83	25.07
Pr16	2514.08	323083.80	2250.40	0.00	4.79	128.51
Pr17	295.18	727641.00	85.60	2.62	257.15	2465.08
Pr18	373.98	115752.60	99.84	0.22	36.44	309.52

Neste experimento é pertinente notar que a utilização de um algoritmo DE no nível do seguidor não afetou em grande medida a resposta do seguidor para

a maior parte dos casos de teste. Apesar disto, há alguns casos de convergência mais complexa (especialmente Pr7 e Pr10) que apresentaram um número elevado de execuções sem sucesso. Observando atentamente estes dois casos, em Pr7 há dificuldade relacionada à não existência de limitantes superiores para as variáveis do líder e do seguidor, sejam estes estipulados direta ou indiretamente, o que induz amplitude exagerada na função do seguidor para valores elevados nas variáveis do líder e torna sua resolução complexa para determinadas configurações. Já no caso Pr10, há de fato uma combinação de restrições que tornam complexo atingir soluções viáveis no problema, soando razoável o comportamento observado.

Outro fator pertinente a ser notado é o número de soluções requeridas pela DE utilizada no nível do seguidor. Aparentemente, para casos de teste de pequena dimensão, trata-se de um número elevado, especialmente quando se compara o número de operações de pivotamento utilizadas pelo método de Lemke nas mesmas soluções. De fato, não é possível comparar em custo operações de pivotamento e o tratamento de uma solução da DE, dadas as operações distintas que os métodos utilizam, custo de geração de números aleatórios na DE, entre outros. Entretanto, é possível imaginar que a utilização da DE nestes casos de teste teve um custo maior quando comparado ao algoritmo de Lemke, especialmente tendo em conta que a menor relação entre estes dois valores foi com número de soluções 25 vezes maior que o número de operações de pivotamento em média e observando que, em sua maior parte, há um número 100 vezes maior de soluções.

Considerando os pontos observados, pode-se dizer que o esforço computacional, somado à garantia de uma resolução exata, depõe a favor do método de Lemke nestas execuções.

Por fim, foi possível verificar o número de iterações médio necessário para execução da DE em cada instância. Excetuando instâncias complexas, em que o DE apresentou dificuldade inclusive em fornecer a resposta correta, pode-se verificar que a maior parte das instâncias exigiu um número reduzido de iterações tendo uma escolha de, por exemplo, 30 iterações, atendido em média à maior parte dos

casos analisados. Apesar disto, instâncias mais complexas, onde foi difícil atingir a resposta correta, necessitaram de um número bem elevado de iterações, especialmente considerando a característica aninhada de uma implementação DE+DE.

Em um segundo experimento, realizou-se a comparação direta entre os algoritmos DE+DE e DE+LEMKE. O intuito foi verificar o desempenho do algoritmo DE proposto para o líder, acompanhado de uma resolução inexata via DE no seguidor, e os impactos desta escolha na convergência da técnica como um todo. Foram novamente realizados 50 experimentos para cada uma das instâncias, tendo como critério de parada para o líder a proximidade relativa do valor de função objetivo da melhor solução encontrada neste nível em 10^{-2} ou 10.000 avaliações de função. Para o seguidor, um critério de parada utilizado na implementação da DE se baseia na proposta de Sinha et al. (2012), onde verifica-se a razão da variância da população na geração G comparada à variância da população inicial dada por

$$\alpha = \sum_i \frac{\sigma^2(x_{i,G})}{\sigma^2(x_{i,init})}$$

tendo como terminação um limiar para este valor; aqui $\alpha < 10^{-5}$. Outra possibilidade de parada é a execução de 1.000 iterações.

A Tabela 6.4 fornece uma compilação dos resultados obtidos nos experimentos apresentando a porcentagem de execuções que atingiram a proximidade da melhor solução da literatura no nível do líder (coluna % Acerto) e o número médio de avaliações da função objetivo do líder até atingir a melhor solução encontrada no processo (coluna Aval. Líder). É informado ainda o número médio de operações de pivotamento realizadas em todo o processo para o algoritmo DE+LEMKE (coluna OP), enquanto para o DE+DE tem-se o correspondente número de soluções geradas (coluna Soluções).

Tendo em vista o baixo número de falhas no geral para resolução do seguidor, verificou-se que não houve grande diferença na convergência geral do método com a escolha de uma implementação da DE ou utilização do método de Lemke no seguidor, observação em linha com o que já foi verificado em outros trabalhos

Tabela 6.4: Comparação DE+LEMKE x DE+DE para instâncias da literatura

Teste	DE+LEMKE			DE+DE		
	% Acerto	Aval. Líder	OP	% Acerto	Aval. Líder	Soluções
Pr1	100	164.58	1324.30	100	160.96	271727.40
Pr2	100	177.54	1501.00	100	166.08	42616828.80
Pr3	92	551.92	3732.98	84	1115.02	38267186.40
Pr4	88	165.28	2709.48	92	853.04	635142.00
Pr5	100	51.82	508.18	100	51.32	2765655.00
Pr6	100	58.08	318.08	100	57.12	107487.00
Pr7	100	89.80	56.70	100	77.70	8229136.20
Pr8	100	24.26	75.24	100	26.38	1689614.40
Pr9	100	198.48	1354.22	100	199.42	587421.00
Pr10	12	1978.70	45574.60	12	5411.02	13851265.20
Pr11	100	582.58	7928.14	100	606.02	132721881.60
Pr12	100	987.42	51484.16	100	1158.06	35485797.00
Pr13	100	245.56	2692.68	100	237.32	2643667.80
Pr14	88	612.30	4758.10	86	941.90	5907429.60
Pr15	100	56.86	389.66	100	72.70	126911.40
Pr16	76	620.34	2913.62	78	1420.58	1790388.60
Pr17	100	67.94	317.08	100	563.72	1832467.20
Pr18	100	87.74	371.60	100	99.66	4494514.80

(Angelo e Barbosa, 2015). O número de execuções de sucesso foi equivalente nas duas opções. Instâncias mais complexas, como Pr10, de fato tiveram um número menor de soluções a convergir ao melhor valor da literatura, mas na maior parte dos problemas foi possível atingir este melhor valor em todas as execuções. Relevante notar também que os casos de teste com grande percentual de execuções sem sucesso na DE do seguidor não tiveram impactados seu percentual de acertos neste experimento, tendo os casos provavelmente restritos à soluções a certa distância do valor ótimo.

Por outro lado, observou-se um crescimento no número de avaliações de função objetivo do líder necessárias. Este comportamento, dado que o percentual de falhas, no geral, foi pequeno, é inesperado mas muito provavelmente se deve ao critério de parada especificado. Como não há nestas execuções um valor alvo a ser atingido, podem ocorrer paradas prematuras na execução da DE que levam a um número maior de respostas incorretas no seguidor. A estipulação de um critério de parada adequado é, de fato, uma dificuldade comum a metaheurísticas e têm seus efeitos intensificados pela necessidade do valor ótimo no BLPP.

6.3.2 Instâncias escaláveis

Para as instâncias escaláveis foram feitos testes nos mesmos moldes das instâncias na literatura, comparando-se a utilização do método de Lemke a uma implementação de Evolução Diferencial no nível de seguidor.

A Tabela 6.5 apresenta os resultados do experimento verificando a capacidade da DE implementada para o seguidor atingir soluções com valor de função objetivo próximo aos obtidos pelo método de Lemke. Desta forma, novamente são apresentados o número médio de operações de pivotamento, excluindo aqui as execuções finalizadas com parada em raio (coluna OP), número médio de soluções geradas (coluna Soluções), número médio de chamadas realizadas à DE no nível do seguidor (coluna Cham.), porcentagem média das execuções da DE do seguidor onde não foi atingida proximidade mínima da solução encontrada via método de Lemke (coluna (%) S/ Sucesso), as médias do número de iterações médio necessário para convergência da DE para o seguidor por chamada realizada desconsiderando os casos de insucesso (coluna Iter./Cham.) e a razão da média do número de soluções criadas pela DE para o seguidor, comparando-o à média do número de operações de pivotamento do método de Lemke, considerando as mesmas configurações fornecidas pelo líder (coluna Sol./OP).

É possível verificar que, diferentemente das instâncias da literatura, não houve nenhum caso de teste onde ocorreu um grande número de insucessos com a execução do DE no nível do seguidor. De fato, as instâncias geradas não tiveram como foco trazer grande dificuldade à convergência no nível de seguidor, mas apresentar relações de certo grau de complexidade entre líder e seguidor. Desta forma, apenas o caso de teste Cquad4, que de fato apresenta um grau maior de complexidade relacionado às restrições do seguidor, teve um número mais elevado de execuções falhas.

Em relação ao número de soluções geradas em comparação às operações de pivotamento, é pertinente notar o número elevado de soluções necessários em praticamente todas as instâncias. Além disto, há um crescimento nesta relação

Tabela 6.5: Comparação DE x Lemke para instâncias escaláveis

Teste	Lemke	DE				Sol./ O.P.
	O.P.	Soluções	Cham.	(%) S/ sucesso	Iter./ Cham.	
SMDquad1(10)	976.20	631330.80	480.16	0.00	43.83	646.72
(20)	3922.60	3300225.60	971.04	0.00	113.29	841.34
(30)	8631.78	7603849.20	1426.74	0.00	177.65	880.91
SMDquad2(10)	931.24	534151.20	465.64	0.01	38.15	573.59
(20)	3725.28	2809482.00	931.34	0.02	100.40	754.17
(30)	8434.66	6863165.40	1405.84	0.01	162.63	813.69
SMDquad3(10)	946.94	631064.40	473.52	0.00	44.42	666.42
(20)	3949.12	3424122.00	987.48	0.00	115.58	867.06
(30)	8744.62	7891345.80	1457.76	0.00	180.44	902.42
SMDquad4(10)	1061.88	491149.20	496.24	0.00	32.99	462.53
(20)	4349.90	2752198.20	1020.88	0.00	89.84	632.70
(30)	9613.40	6715757.40	1513.70	0.00	147.88	698.58
SMDquad5(10)	2808.80	797535.60	561.78	0.00	47.32	283.94
(20)	10926.66	4083045.60	1092.72	0.00	124.55	373.68
(30)	24035.84	9523863.60	1602.48	0.00	198.11	396.24
SMD6(10)	1150.32	464278.20	575.16	0.00	26.91	403.61
(20)	4391.50	2843790.00	1097.90	0.00	86.34	647.57
(30)	9596.80	8122530.60	1599.52	0.00	169.27	846.38
Cquad1(10)	60065.56	5455404.60	9435.90	0.00	19.27	90.82
(20)	93859.30	19960973.40	10008.14	0.00	66.47	212.67
(30)	122243.14	36552048.60	10008.88	0.00	121.72	299.01
Cquad2(10)	5324.88	1134713.40	887.48	0.00	42.62	213.10
(20)	22176.44	5987338.80	2016.04	0.00	98.99	269.99
(30)	43843.20	12801157.80	2740.20	0.00	155.71	291.98
Cquad3(10)	48139.40	11991589.20	4829.42	0.00	82.77	249.10
(20)	154676.52	54011539.20	7744.60	0.00	232.46	349.19
(30)	284699.31	111408710.20	9497.94	0.00	390.96	391.32
Cquad4(10)	35320.76	21194124.60	2243.30	1.73	297.65	600.05
(20)	139063.96	73726068.60	4473.18	1.13	538.08	530.16
(30)	271594.40	144000738.00	5643.64	1.00	840.53	530.21

com o aumento da dimensão em quase todos os casos de teste, à exceção do caso Cquad4 onde há diminuição neste valor.

Este comportamento inverso presente no caso Cquad4, aliado ainda a um decréscimo no número de execuções falhas, é um aspecto curioso podendo se dever à maior dificuldade no processo de convergência com o aumento de dimensão, mantendo as soluções por mais tempo em uma região que demanda uma menor dificuldade de processamento no DE.

Também foi realizado o experimento envolvendo a comparação de desempenho entre as técnicas DE+LEMKE e DE+DE para as instâncias escaláveis. Do mesmo modo como nas instâncias da literatura, a Tabela 6.6 apresenta a porcentagem de execuções que atingiram a proximidade da melhor solução da literatura no nível do líder (coluna % Acerto) e o número médio de avaliações da função objetivo do líder até atingir a melhor solução encontrada no processo (coluna Aval. Líder). É informado ainda o número médio de operações de pivotamento realizado em todo o processo para o algoritmo DE+LEMKE (coluna OP), enquanto para o DE+DE tem-se o correspondente número de soluções geradas (coluna Soluções).

Tabela 6.6: Comparação DE+LEMKE x DE+DE para instâncias escaláveis

Teste	DE+LEMKE			DE+DE		
	% Acerto	Aval. Líder	OP	% Acerto	Aval. Líder	Soluções
SMDquad1(10)	100.00	487.98	1100.40	100.00	478.56	1429271.40
(20)	100.00	951.66	4376.32	100.00	945.46	6562418.40
(30)	100.00	1413.06	9985.38	100.00	1407.10	15079900.20
SMDquad2(10)	100.00	457.60	1039.26	100.00	451.52	4313430.00
(20)	100.00	925.00	4324.20	100.00	975.22	14076072.00
(30)	100.00	1413.50	10138.66	100.00	1348.10	32126310.00
SMDquad3(10)	100.00	468.66	1085.40	100.00	474.52	27306117.00
(20)	100.00	976.32	4898.64	100.00	979.64	80302458.60
(30)	100.00	1464.14	11768.54	100.00	1501.48	149610004.20
SMDquad4(10)	100.00	494.74	1347.86	100.00	492.64	2480647.20
(20)	100.00	986.42	5728.22	100.00	1004.18	9893071.80
(30)	100.00	1507.78	13785.14	100.00	1508.82	23378061.60
SMDquad5(10)	100.00	550.50	3504.08	100.00	538.94	44378205.60
(20)	100.00	1080.84	15005.84	100.00	1091.88	122006846.40
(30)	100.00	1638.28	35775.84	100.00	1581.88	220133151.00
SMD6(10)	100.00	553.76	1207.04	0.00	8061.38	2142541885.20
(20)	100.00	1082.92	4805.24	0.00	5962.06	284434494.00
(30)	100.00	1564.38	10673.10	0.00	8654.14	139361669.40
Cquad1(10)	8.00	3160.96	81002.70	4.00	5577.46	23925246.00
(20)	0.00	5474.88	165553.82	0.00	6485.76	334311813.60
(30)	0.00	7896.48	258957.12	0.00	6022.94	195764561.40
Cquad2(10)	100.00	933.66	11820.20	100.00	831.06	9558987.00
(20)	100.00	2088.06	53244.40	100.00	1999.22	80500148.40
(30)	100.00	2870.94	111398.40	100.00	2688.96	289094148.60
Cquad3(10)	50.00	2563.76	80547.62	70.00	2235.96	34415517.00
(20)	22.00	6240.06	339135.22	12.00	6338.49	211597359.80
(30)	8.00	9358.58	749657.64	6.00	7298.18	433437115.80
Cquad4(10)	90.00	2182.78	47456.44	98.00	1808.84	10828067.40
(20)	80.00	4595.62	183161.08	90.00	3698.98	71477889.00
(30)	80.00	5591.60	387365.88	68.00	5781.16	202490436.60

Semelhante ao que foi verificado nos casos da literatura, não houve, para a maioria dos casos de teste, diferença notável no percentual de acertos dos testes

realizados. Os casos mais complexos, Cquad1 e Cquad3, apresentaram um baixo número de execuções com sucesso em ambas as abordagens, enquanto o caso mais simples Cquad2 não proporcionou maiores dificuldades às técnicas, assim como a grande maioria dos casos apenas com restrições de “caixa” SMDquad.

A exceção se dá pela instância SMD6, vinda da literatura e sem modificações. Nesta, enquanto a abordagem com o método de Lemke resolve com relativa facilidade, a utilização de um DE no nível do seguidor não é capaz de, em nenhum momento, atingir o valor ótimo. Além disto, há um comportamento estranho no crescimento do número de soluções geradas pela abordagem DE+DE e número de avaliações do líder, com o caso intermediário de 20 dimensões apresentando um menor valor nestas quantidades que a versão com 10 dimensões. Estas duas incongruências aparentes podem ser explicadas pela existência de múltiplos ótimos no nível do seguidor para este caso. Desta forma, como não foi especificada na implementação uma abordagem otimista ou pessimista, não há uma pressão seletiva em direção ao melhor ótimo para o líder, dificultando sensivelmente a convergência para o ótimo da literatura, que leva em conta uma abordagem otimista. Além disto, a existência de múltiplos ótimos leva a uma maior dificuldade na convergência de uma população, fator que pode ser agravado em menores dimensões, fazendo com que um critério de parada como a variância não seja atingido para um número grande de execuções e aumentando o número de soluções geradas na DE do seguidor em dimensões menores. Em relação ao número de avaliações do líder divergente, uma possibilidade pode ser a grande dificuldade em se obter melhores soluções quando a resposta do seguidor não é única, fazendo com que o processo evolutivo se estagne cedo em instâncias mais complexas, o que economizaria avaliações em proporção maior quando comparado ao impacto do aumento de dimensão entre 10 e 20 variáveis.

Há ainda outro resultado estranho relativo ao caso de teste Cquad1, onde a utilização de 20 dimensões apresentou número muito superior de soluções criadas quando comparado ao caso de 10 ou 30 dimensões. Apesar de se destacar à primeira

vista, este se deve a uma única execução com número muito elevado de soluções criadas, tendo as demais o comportamento esperado, com média de 89022461.02.

Em relação ao número de avaliações no nível superior para os demais casos, a quantidade necessária para se atingir a melhor solução é equivalente em ambas as abordagens, o que, como comentado anteriormente, provavelmente se deve ao pouco foco posto em atribuir complexidade à resolução do nível do seguidor e consequente bom desempenho da técnica DE, tornando-a equivalente, neste ponto de vista, à utilização do método de Lemke.

6.4 Desempenho global da técnica DE+LEMKE

Em um terceiro conjunto de experimentos, buscou-se verificar o desempenho global da técnica DE+LEMKE em termos de qualidade de solução atingida e robustez destes resultados. Para isso foram realizadas novamente 50 execuções para todas as instâncias, coletando a melhor solução fornecida pelo processo. O critério de parada atribuído aos algoritmos avaliados foi de 6.000 avaliações de função para o líder ou 10.000 iterações do algoritmo DE neste nível.

Como ponto de comparação utilizou-se o algoritmo implementado para o experimento anterior DE+DE, com inclusão do mesmo critério de parada baseado na variância da população para o seguidor, tendo como limite $\alpha < 10^{-5}$ ou 1.000 iterações.

Ainda, como foi possível verificar que a melhor solução obtida pelo processo evolutivo é encontrada prematuramente mesmo em execuções onde não se atinge a melhor solução conhecida, propõe-se aqui uma modificação do algoritmo DE+LEMKE combinando este a uma técnica simples de reinicialização da população.

Denominado R-DE+LEMKE, nesta variante é aplicado um critério de reinicialização do processo evolutivo baseado na razão de variância da população em uma geração G em relação à população inicial (mesmo critério aplicado como parada para o seguidor na implementação DE+DE), utilizando novamente o limite

$\alpha < 10^{-5}$. Desta forma, toda vez que este critério é atendido, tem-se a eliminação de toda a população corrente, substituída por uma nova gerada de maneira aleatória. Mantém-se ainda na execução a melhor solução encontrada até o momento, sendo esta aproveitada na operação de mutação como elemento *best*.

A seguir, a Tabela 6.7 apresenta a melhor solução obtida dentre as 50 execuções por cada um dos algoritmos analisados, bem como o ponto relativo a esta solução, valor obtido para o seguidor e a referência da literatura. Esta comparação permite verificar o quão competente são os algoritmos em questão, em especial a técnica DE+LEMKE, tendo em vista o estado da arte.

Teste	Estratégia	(x^*, y^*)	F	f
Pr1	DE+LEMKE	(20.0003, 4.9998, 10, 4.9998)	224.996	100.006
	R-DE+LEMKE	(20.0002, 4.9999, 10, 4.9999)	224.997	100.004
	DE+DE	(20.0001, 5, 10, 4.99678)	224.934	100.001
	Literatura	(20, 5, 10, 5)	225.000	100.000
Pr2 ^a	DE+LEMKE	(2, 2.38896e-17, 1.5, -5e-06)	3.250	4.000
	R-DE+LEMKE	(1.99999, -1.529e-06, 1.49998, 7.759e-06)	3.250	4.000
	DE+DE	(2.00002, 7.9501e-06, 0, 0)	4.000	-2.000
	Literatura	(2, 0, 1.5, 0)	3.250	4.000
Pr3	DE+LEMKE	(-0.0001, 30, -10, 10)	0.000	100.002
	R-DE+LEMKE	(-2.04562e-05, 30.0001, -10, 10)	0.000	100.000
	DE+DE	(7.66638e-05, 29.9887, -10, 9.99433)	-0.006	99.999
	Literatura	(0, 0, -10, -10)	0.000	200.000
Pr4	DE+LEMKE	(0.999999, -3.33333e-06)	17.000	1.000
	R-DE+LEMKE	(1, 1.4656e-05)	17.000	1.000
	DE+DE	(1.0634, 0)	16.497	1.000
	Literatura	(1, 0)	17.000	1.000
Pr5	DE+LEMKE	(0.00522166, 2.00004, 1.875, 0.906259)	-12.679	-1.016
	R-DE+LEMKE	(0.0004509, 2.00005, 1.875, 0.906262)	-12.679	-1.016
	DE+DE	(0.0311547, 1.99914, 1.88609, 0.914373)	-12.707	-1.013
	Literatura	(0, 1.99996, 1.87549, 0.906606)	-12.679	-1.016
Pr6	DE+LEMKE	(1.88889, 0.888887, 0)	-1.210	7.617
	R-DE+LEMKE	(1.88889, 0.888889, 0)	-1.210	7.617
	DE+DE	(1.88826, 0.888364, 0)	-1.211	7.620
	Literatura	(1.88889, 0.888889, 0)	-1.210	7.610

Pr7	DE+LEMKE	(1, 0)	1.000	0.000
	R-DE+LEMKE	(0.999904, 0)	1.000	0.000
	DE+DE	(1, 0)	1.000	0.000
	Literatura	(1, 0)	1.000	0.000
Pr8	DE+LEMKE	(1, 3)	5.000	4.000
	R-DE+LEMKE	(1.00004, 3.00009)	5.000	4.000
	DE+DE	(0.990909, 2.97555)	4.988	4.098
	Literatura	(1, 3)	5.000	4.000
Pr9	DE+LEMKE	(-0.0001, 0.90002, 0, 0.60008, 0.39992)	-29.202	3.200
	R-DE+LEMKE	(-9.988e-05, 0.90001, 0, 0.60007, 0.3999)	-29.202	3.200
	DE+DE	(-9.008e-05, 0.899886, 0.00026, 0.600242, 0.399924)	-29.207	3.200
	Literatura	(0, 0.9, 0, 0.6, 0.4)	-29.200	3.200
Pr10	DE+LEMKE	(0.4995, 0.8001, 0, 0.2004, 0.7996)	-18.411	1.800
	R-DE+LEMKE	(0.499515, 0.800095, 0, 0.200387, 0.799608)	-18.410	1.800
	DE+DE	(0.479759, 0.803189, 0, 0.213485, 0.786049)	-18.734	1.786
	Literatura	(0.5, 0.8, 0, 0.2, 0.8)	-18.400	1.800
Pr11	DE+LEMKE	(1.5456, 0.77989, 0.156385, 2.20985, 1.88727, 0)	14.989	-16.986
	R-DE+LEMKE	(1.54551, 0.780074, 0.156246, 2.20984, 1.8874, 0)	14.992	-16.987
	DE+DE	(1.54498, 0.779319, 0.156163, 2.21007, 1.88675, 0)	14.982	-16.981
	Literatura	(1.55, 0.78, 0.16, 2.21, 1.89, 0)	14.990	-16.990
Pr12	DE+LEMKE	(0.0004, 9.8757, 9.9999, 0.0024, 7.0326, 4.2442, 0.0001, 9.9998, 0.0005, 9.9990, 3.098, 10, 10, 10, 0, 9.9986)	-466.827	-10.714
	R-DE+LEMKE	(0.0004, 9.8757, 9.9999, 0.0024, 7.0326, 4.2442, 0.0001, 9.9998, 0.0005, 9.9990, 3.098, 10, 10, 10, 0, 9.9986)	-466.827	-10.714
	DE+DE	(4.87e-05, 7.2510, 9.9999, 9.05e-05, 8.4700, 2.7748, -2.20e-05, 9.9999, 0.0060, 9.9995, 3.1823, 10, 10, 10, 0, 9.9977)	-467.569	-10.467
	Literatura	(0, 2.44, 10, 0, 10, 8.74, 5.25, 10, 0, 10, 3.73, 10, 10, 10, 0, 0)	-453.610	-68.810

Pr13 ^a	DE+LEMKE	(7.35587, 3.53376, 11.6442, 17.4663, 0.889575, 10.0001, 29.1105, 0)	6600.010	18.67, 38.69
	R-DE+LEMKE	(7.37148, 3.55758, 11.6284, 17.4426, 0.929533, 9.99952, 29.071, 0)	6600.010	18.43, 39.15
	DE+DE	(7.92998, 4.39059, 11.0718, 16.6076, 2.3147, 10.0059, 27.6795, 0)	6600.020	11.81, 57.59
	Literatura	(7.02, 3.03, 11.98, 17.97, 0.05, 10.00, 29.95, 0)	6600.000	24.58, 29.54
Pr14	DE+LEMKE	(-6.35905e-05, 30.0003, -10, 10.0001)	0.000	100.001
	R-DE+LEMKE	(3.26149e-05, 29.9999, -10, 9.99992)	0.000	99.999
	DE+DE	(-6.15332e-05, 30.0005, -10, 10.0003)	0.000	100.001
	Literatura	(0.00, 30.00, -10.00, 10.00)	0.000	100.000
Pr15	DE+LEMKE	(1.88889, 0.888889, 0)	0.000	7.617
	R-DE+LEMKE	(1.88889, 0.888889, 0)	0.000	7.617
	DE+DE	(1.88879, 0.888966, 0)	0.000	7.617
	Literatura	(1.89, 0.89, 0.00)	0.000	7.610
Pr16	DE+LEMKE	(-0.400004, 0.800008, 2.00002, 0)	-3.920	-2.000
	R-DE+LEMKE	(-0.397873, 0.801089, 2, 0)	-3.920	-2.000
	DE+DE	(-0.397715, 0.801005, 2.00009, 0)	-3.920	-1.999
	Literatura	(-0.4050, 0.7975, 2, 0)	-3.920	-2.000
Pr17	DE+LEMKE	(2.85634, 3.88075, 3.04016)	0.849	-22.951
	R-DE+LEMKE	(2.8562, 3.88065, 3.04007)	0.849	-22.949
	DE+DE	(2.92826, 2.81754, 2.92768)	0.592	-19.769
	Literatura	(2.8563, 3.8807, 3.0402)	0.849	-22.951
Pr18	DE+LEMKE	(1.90953, 2.9786, 2.23216)	1.563	-11.683
	R-DE+LEMKE	(1.90953, 2.9786, 2.23216)	1.563	-11.683
	DE+DE	(1.90575, 2.97909, 2.23381)	1.560	-11.663
	Literatura	(1.9095, 2.9786, 2.2321)	1.563	-11.683

Tabela 6.7: Comparação das melhores solução obtidas nas técnicas DE+LEMKE, R-DE+LEMKE e DE+DE em instâncias da literatura. O superscrito “a” indica um problema de maximização em ambos os níveis.

Ao se analisar os melhores valores obtidos, as três abordagens apresentam desempenho em linha com o que é observado na literatura para a maior parte dos casos de teste. Ainda, no problema Pr12 foi possível encontrar um novo ótimo, com valor inferior ao observado em trabalhos na literatura para essa instância,

indicando a eficácia da técnica. Por outro lado, mesmo apresentando desempenho semelhante à utilização do método de Lemke, a escolha da utilização de uma DE no problema do seguidor levou à informação incorreta nos casos de teste Pr2 e Pr17, indicando um revés para esta opção.

Neste mesmo conjunto de experimentos a Tabela 6.8 apresenta os valores da melhor e pior solução obtida, bem como média e mediana das soluções para a função objetivo no nível do líder, considerando as instâncias da literatura.

Teste	Estratégia	Melhor	Média	Mediana	Pior
Pr1	DE+LEMKE	225.00	225.00	225.00	225.00
	R-DE+LEMKE	225.00	225.00	225.00	225.01
	DE+DE	224.93	224.96	224.96	224.98
Pr2 ^a	DE+LEMKE	3.25	3.25	3.25	3.25
	R-DE+LEMKE	3.25	3.25	3.25	3.25
	DE+DE	4.00	3.30	3.25	3.25
Pr3	DE+LEMKE	0.00	0.40	0.00	5.00
	R-DE+LEMKE	0.00	0.00	0.00	0.01
	DE+DE	-0.01	0.79	0.00	4.97
Pr4	DE+LEMKE	17.00	17.96	17.00	25.00
	R-DE+LEMKE	17.00	17.00	17.00	17.00
	DE+DE	16.50	17.26	16.60	24.98
Pr5	DE+LEMKE	-12.68	-12.68	-12.68	-12.65
	R-DE+LEMKE	-12.68	-12.68	-12.68	-12.68
	DE+DE	-12.71	-12.69	-12.69	-12.67
Pr6	DE+LEMKE	-1.21	-1.21	-1.21	-1.21
	R-DE+LEMKE	-1.21	-1.21	-1.21	-1.21
	DE+DE	-1.21	-1.21	-1.21	-1.21
Pr7	DE+LEMKE	1.00	1.00	1.00	1.00
	R-DE+LEMKE	1.00	1.00	1.00	1.00
	DE+DE	1.00	1.00	1.00	1.00
Pr8	DE+LEMKE	5.00	5.00	5.00	5.00
	R-DE+LEMKE	5.00	5.00	5.00	5.00
	DE+DE	4.99	4.99	4.99	4.99

Pr9	DE+LEMKE	-29.20	-29.20	-29.20	-29.20
	R-DE+LEMKE	-29.20	-29.20	-29.20	-29.20
	DE+DE	-29.21	-29.20	-29.20	-29.20
Pr10	DE+LEMKE	-18.41	-16.29	-16.00	-16.00
	R-DE+LEMKE	-18.41	-16.87	-16.00	-16.00
	DE+DE	-18.73	-16.35	-16.03	-16.01
Pr11	DE+LEMKE	14.99	14.99	14.99	14.99
	R-DE+LEMKE	14.99	15.00	15.00	15.00
	DE+DE	14.98	14.99	14.99	14.99
Pr12	DE+LEMKE	-466.83	-454.38	-453.61	-453.41
	R-DE+LEMKE	-466.83	-454.35	-453.58	-453.56
	DE+DE	-467.57	-453.91	-453.56	-448.06
Pr13 ^a	DE+LEMKE	6600.01	6600.01	6600.01	6600.01
	R-DE+LEMKE	6600.01	6600.01	6600.01	6600.01
	DE+DE	6600.02	6599.73	6599.87	6595.86
Pr14	DE+LEMKE	0.00	0.60	0.00	5.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.70	0.00	4.97
Pr15	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00
Pr16	DE+LEMKE	-3.92	-3.89	-3.92	-3.79
	R-DE+LEMKE	-3.92	-3.92	-3.92	-3.92
	DE+DE	-3.92	-3.89	-3.92	-3.79
Pr17	DE+LEMKE	0.85	0.85	0.85	0.85
	R-DE+LEMKE	0.85	0.85	0.85	0.85
	DE+DE	0.59	0.74	0.75	0.77
Pr18	DE+LEMKE	1.56	1.56	1.56	1.56
	R-DE+LEMKE	1.56	1.56	1.56	1.56
	DE+DE	1.56	1.56	1.56	1.56

Tabela 6.8: Comparação dos resultados obtidos pelas técnicas DE+LEMKE, R-DE+LEMKE e DE+DE em instâncias da literatura. O superscrito “a” indica um problema de maximização em ambos os níveis.

Nesta tabela é possível observar que em aproximadamente metade dos casos de teste o desempenho é semelhante entre os algoritmos e se aproxima em todas as execuções da melhor solução conhecida. Para os casos onde isto não ocorre, pode-se observar que a estratégia DE+LEMKE apesar de conseguir encontrar o ótimo conhecido ao menos uma vez, apresenta dificuldades para atingir este ponto em todas as execuções realizadas. Esta ocorrência já se apresenta mais rara no R-DE+LEMKE, onde apenas na instância Pr10 surgem execuções em que não se atinge este ótimo. O algoritmo DE+DE apresenta dificuldade semelhante ao DE+LEMKE relacionada à convergência em algumas execuções, mas possui também, como observado anteriormente, outro problema relativo à descoberta de falsos ótimos, notadamente nos casos de teste Pr2 e Pr17. Para o caso de teste Pr2 a ocorrência pode não ser considerada tão grave dado que a mediana apresenta uma solução com valor correto, indicando que, apesar de alguns deslizes, o algoritmo DE+DE tem um desempenho aceitável. Mas o mesmo não pode ser dito do caso de teste Pr17, visto que todas as execuções encontram um falso ótimo, tendo o DE+DE grande dificuldade em obter uma solução final correta.

Da mesma forma que apresentado para as instâncias da literatura, a Tabela 6.9 exibe os valores da melhor e pior solução obtida, bem como média e mediana das soluções para a função objetivo no nível do líder considerando as instâncias escaláveis.

Teste	Estratégia	Melhor	Média	Mediana	Pior
SMDquad1(10)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00
SMDquad1(20)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00
SMDquad1(30)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00

SMDquad2(10)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	-1.07	-0.06	0.00	0.00
SMDquad2(20)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	-1.01	-0.36	0.00	0.00
SMDquad2(30)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	-1.28	-0.35	0.00	0.00
SMDquad3(10)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00
SMDquad3(20)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00
SMDquad3(30)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00
SMDquad4(10)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00
SMDquad4(20)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	-1.00	-0.06	0.00	0.00
SMDquad4(30)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	-1.00	-0.08	0.00	0.00
SMDquad5(10)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00
SMDquad5(20)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00
SMDquad5(30)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.00	0.00	0.00	0.00

SMD6(10)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	0.04	0.24	0.21	0.52
SMD6(20)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	2.06	11.00	10.73	24.39
SMD6(30)	DE+LEMKE	0.00	0.00	0.00	0.00
	R-DE+LEMKE	0.00	0.00	0.00	0.00
	DE+DE	2.51	11.55	10.76	24.14
Cquad1(10)	DE+LEMKE	-1.00	-0.62	-0.60	-0.20
	R-DE+LEMKE	-1.00	-0.94	-1.00	-0.60
	DE+DE	-1.00	-0.65	-0.60	-0.20
Cquad1(20)	DE+LEMKE	-0.70	-0.35	-0.30	-0.10
	R-DE+LEMKE	-0.80	-0.59	-0.60	-0.40
	DE+DE	-0.70	-0.40	-0.40	0.14
Cquad1(30)	DE+LEMKE	-0.40	-0.24	-0.27	-0.04
	R-DE+LEMKE	-0.59	-0.41	-0.40	-0.20
	DE+DE	-0.53	-0.24	-0.25	0.08
Cquad2(10)	DE+LEMKE	-2.00	-2.00	-2.00	-2.00
	R-DE+LEMKE	-2.00	-2.00	-2.00	-2.00
	DE+DE	-2.00	-2.00	-2.00	-2.00
Cquad2(20)	DE+LEMKE	-2.00	-2.00	-2.00	-1.99
	R-DE+LEMKE	-2.00	-2.00	-2.00	-1.99
	DE+DE	-2.00	-2.00	-2.00	-1.98
Cquad2(30)	DE+LEMKE	-2.00	-1.99	-2.00	-1.94
	R-DE+LEMKE	-2.00	-1.99	-2.00	-1.94
	DE+DE	-2.00	-2.00	-2.00	-1.98
Cquad3(10)	DE+LEMKE	-14.00	-6.50	-6.50	1.00
	R-DE+LEMKE	-14.00	-6.78	-13.72	1.00
	DE+DE	-14.23	-9.64	-14.20	1.00
Cquad3(20)	DE+LEMKE	-14.00	-2.30	1.00	1.00
	R-DE+LEMKE	-14.00	-2.29	1.00	1.01
	DE+DE	-20.56	-0.99	1.00	1.00
Cquad3(30)	DE+LEMKE	-14.00	-0.20	1.00	1.00
	R-DE+LEMKE	-13.98	-0.19	1.01	1.01
	DE+DE	-14.15	-0.19	1.00	1.00

Cquad4(10)	DE+LEMKE	1.00	1.01	1.00	1.10
	R-DE+LEMKE	1.00	1.01	1.01	1.11
	DE+DE	1.00	1.00	1.00	1.10
Cquad4(20)	DE+LEMKE	1.00	1.01	1.00	1.05
	R-DE+LEMKE	1.01	1.02	1.01	1.07
	DE+DE	1.00	1.01	1.00	1.10
Cquad4(30)	DE+LEMKE	1.00	1.01	1.00	1.10
	R-DE+LEMKE	1.01	1.02	1.02	1.12
	DE+DE	1.00	1.02	1.01	1.12

Tabela 6.9: Comparação dos resultados obtidos pelas técnicas DE+LEMKE, R-DE+LEMKE e DE+DE em instâncias escaláveis

Nesta tabela é possível constatar que se mantém a relação observada entre as estratégias DE+LEMKE e R-DE+LEMKE, com a segunda apresentando, quase sempre, melhor valor em média e, no caso da instância Cquad1, atingindo até a melhor solução geral na comparação. Apesar disto, é pertinente notar que em algumas instâncias, especialmente Cquad4, a técnica R-DE+LEMKE apresenta desempenho ligeiramente inferior, o que muito provavelmente se deve ao critério de reinício não totalmente ajustado, impedindo a convergência completa da técnica. Em relação à técnica DE+DE, observa-se o mesmo problema das instâncias da literatura, havendo execuções com valor inferior ao ótimo. Entretanto, um ponto relevante é que apenas no caso de teste Cquad3 com 10 dimensões a mediana também é comprometida por esta falha na resolução, sendo os demais casos em execuções pontuais. O DE+DE ainda apresenta dificuldade na resolução do caso de teste SMD6 pela, já comentada, existência de múltiplas soluções do seguidor para a configuração ótima do líder. Nos demais casos, o DE+DE se apresenta em linha com o desempenho da técnica DE+LEMKE, não havendo grandes destaques.

Por fim, a Figura 6.8 exibe um gráfico de Perfil de Desempenho (Dolan e Moré, 2002) compreendendo as execuções realizadas para as instâncias escaláveis. A visualização via Perfil de Desempenho apresenta o comportamento da função de probabilidade acumulada $\rho(\tau)$ para uma dada técnica em um conjunto de casos teste, onde τ é a razão entre o desempenho da técnica em questão para uma instância analisada quando com-

parada ao melhor desempenho observado nesta instância. Este formato permite não só analisar as técnicas que apresentam melhor desempenho, mas também verificar aquelas que se apresentam mais consistentes, tendo um desempenho próximo ao melhor em um grande número de instâncias.

No Perfil de Desempenho apresentado foi utilizada como métrica de desempenho a diferença entre o valor médio da função objetivo do problema no nível do líder obtido pelas execuções e o valor referente à melhor solução conhecida. Foram ainda removidas as execuções da técnica DE+DE onde tem-se soluções com valor inferior ao mínimo possível visando uma comparação mais correta de desempenho entre as estratégias.

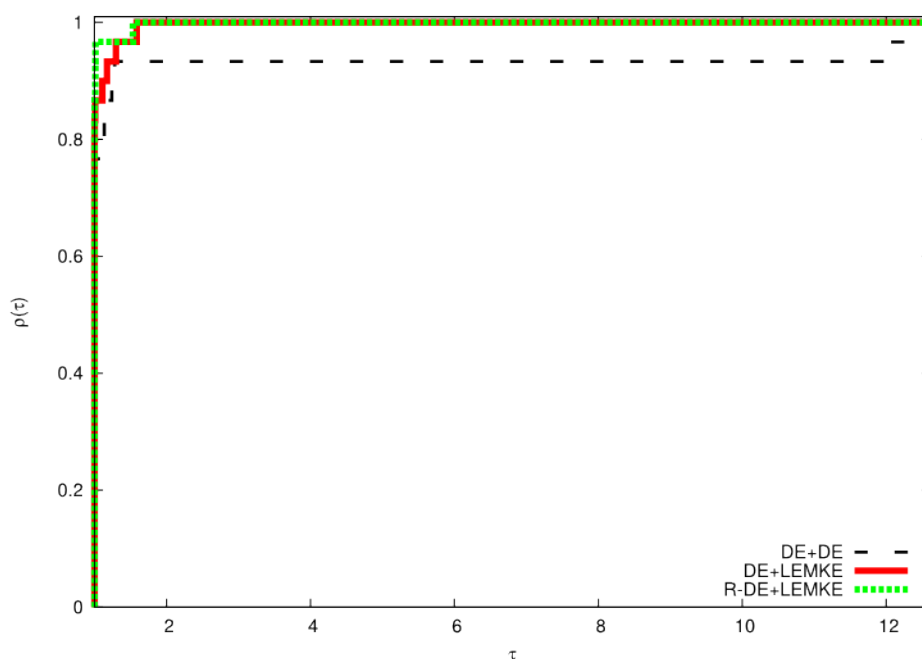


Figura 6.8: Perfil de Desempenho das técnicas DE+LEMKE, R-DE+LEMKE e DE+DE em instâncias escaláveis

É possível observar que a técnica que apresentou melhor desempenho ($\tau = 1$) de fato foi a R-DE+LEMKE, corroborando com a análise anterior, tendo a DE+LEMKE um desempenho intermediário e a DE+DE a pior dentre as técnicas analisadas, o que muito se deve ao mal desempenho no caso de teste SMD6.

Verificando a consistência do algoritmo à medida que se aumenta o valor de τ tem-se a estratégia DE+LEMKE encontrando a melhor técnica R-DE+LEMKE em torno do valor $\tau = 2$, demonstrando que não há uma discrepância tão grande entre o desempenho

das duas técnicas. Já a DE+DE apenas é capaz de atingir $\rho(\tau) = 1.0$ para $\tau = 12$, mas tem bom patamar em torno de $\tau = 2$.

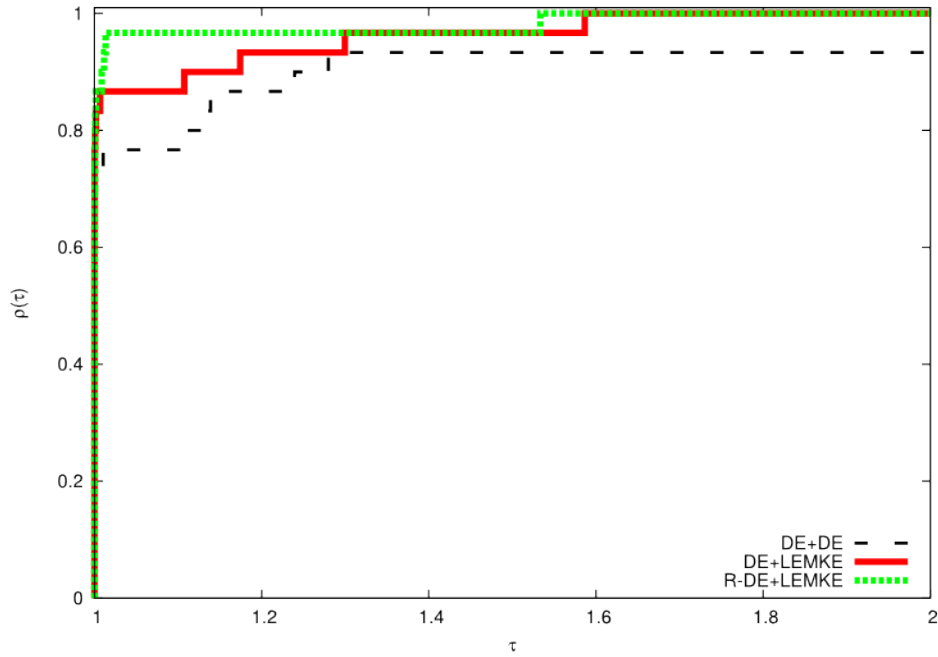


Figura 6.9: Perfil de Desempenho das técnicas DE+LEMKE, R-DE+LEMKE e DE+DE em instâncias escaláveis ($\tau \in [1, 2]$)

Buscando ainda facilitar a visualização a Figura 6.9 trás o Perfil de Desempenho anterior limitando o valor de τ ao intervalo $[1, 2]$. Nesta ficam mais claras as análises feitas anteriormente. O algoritmo R-DE+LEMKE apresenta o melhor desempenho e atinge $\rho = 1$ próximo a $\tau = 1.6$. O DE+LEMKE apresenta desempenho intermediário e é capaz de alcançar o desempenho do melhor algoritmo em torno de $\tau = 1.3$. Enquanto a técnica DE+DE, atinge o patamar próximo a 0.9 em $\tau = 1.3$, com o desempenho na instância SMD6 acarretando esta resistência para se alcançar as outras técnicas.

Capítulo 7

Conclusão e Trabalhos Futuros

O problema de programação em dois níveis é relevante em diversos aspectos teóricos e práticos, porém apresenta grande complexidade na proposição de algoritmos eficazes.

Mesmo com essa complexidade, foi possível notar boas possibilidades ao se considerar instâncias com aspecto mais “amigável”, especialmente no seguidor. Sob certas condições, a formulação do problema do seguidor como um Problema de Complementaridade Linear se mostrou uma boa alternativa dada a existência de métodos eficientes voltados à sua resolução.

Prosseguindo neste sentido, a utilização do algoritmo de Lemke foi uma alternativa que se mostrou viável, tanto ao se verificar as garantias teóricas de convergência, quanto na aplicação aos casos de teste, onde se observou boa capacidade de convergência e com custo médio (representado pelo número de operações) de crescimento controlado, mesmo considerando instâncias mais complexas e com maior número de dimensões.

Ainda, acerca do esforço computacional, não é possível, pela variedade de implementações da Evolução Diferencial e pela existência de variações do algoritmo de Lemke, apresentar estudo definitivo sobre qual destas opções apresenta melhor relação de custo. Mas, as análises aqui realizadas levam a crer que a opção de utilização do algoritmo de Lemke é mais vantajosa, dada a diferença em ordem de grandeza de operações de pivotamento e número de soluções, ao menos na

dimensão de problemas aqui verificada. Esta análise, de fato, não leva em conta questões relacionadas ao escopo do algoritmo, dado que a opção por Lemke limita os casos de aplicação de uma abordagem que a implementa, sendo outra questão a ser verificada.

A abordagem proposta DE+LEMKE se mostrou com bom desempenho, sendo capaz de resolver quase todas as instâncias selecionados da literatura com perfeição, o que serviu no sentido de validá-la como opção viável. Ao se considerar casos de teste escaláveis, foi possível observar também que instâncias apenas com restrições de “caixa” não apresentaram maiores dificuldades, mesmo considerando os casos com 30 dimensões, o que se deve, além do bom desempenho da DE, ao formato mais simplificado destas instâncias se for garantida uma resposta correta do seguidor.

Para as instâncias escaláveis com restrições funcionais, por outro lado, pode-se observar que a estratégia tem limitações, especialmente nos testes com 20 ou 30 dimensões.

Ainda, durante a análise de desempenho, foi verificada uma possibilidade promissora no estabelecimento de critérios de reinício do processo evolutivo do DE utilizado no nível do líder. Esta variante, denominada R-DE+LEMKE, permitiu uma maior robustez nos resultados, para a mesma disponibilidade de recursos computacionais da versão original DE+LEMKE.

Em se tratando de trabalhos futuros, uma possibilidade no contexto da formulação via LCP corresponde à implementação de variantes mais robustas do algoritmo de Lemke, verificando possíveis ganhos de desempenho em sua utilização. Outra possibilidade seria a elaboração de modificações neste método com o intuito de possibilitar a utilização de critérios de escolha otimistas/pessimistas no BLPP. Ainda neste escopo está a resolução do LCP através de outros algoritmos, em especial métodos iterativos capazes de possibilitar parada antecipada (com uma solução aproximada) no intuito de se economizar recursos em soluções iniciais ou menos promissoras.

No contexto de abordagem, a combinação de outras metaheurísticas com o algoritmo de Lemke no seguidor pode ser boa possibilidade, sendo combinação pouco explorada na literatura. Além deste, considerando a observação pontual da abordagem R-DE+LEMKE, é pertinente verificar a possibilidade de se estabelecer outros critérios de reinicialização do processo evolutivo, bem como aproveitamento de conhecimento prévio destas execuções em processos posteriores da evolução. Por fim, a comparação direta desta proposta com outras técnicas já presentes na literatura pode se apresentar como um bom acréscimo às análises de desempenho aqui realizadas, especialmente considerando os novos casos de teste propostos.

Referências Bibliográficas

- E. Aiyoshi e K. Shimizu. A solution method for the static constrained Stackelberg problem via penalty method. **IEEE Trans. on Automatic Control**, 29(12): 1111–1114, 1984.
- G. Anandalingam e D. J. White. A Solution Method for the Linear Static Stackelberg Problem Using Penalty Functions. **IEEE Trans. on Automatic Control**, 35(10):1170–1173, 1990.
- J. S. Angelo e H. J. C. Barbosa. A study on the use of heuristics to solve a bilevel programming problem. **Intl. Trans. in Operational Research**, 22(5): 861–882, sep 2015.
- J. S. Angelo, E. Krempser, e H. J. C. Barbosa. Differential evolution for bilevel programming. In: **2013 IEEE Congress on Evolutionary Computation**, volume 1, páginas 470–477, Cancun, jun 2013. IEEE.
- J. F. Bard. Convex two-level optimization. **Mathematical Programming**, 40-40 (1-3):15–27, jan 1988.
- J. F. Bard. **Practical Bilevel Optimization**, volume 30 of **Nonconvex Optimization and Its Applications**. Springer US, Boston, MA, 1998.
- J. F. Bard e J. E. Falk. An explicit solution to the multi-level programming problem. **Computers and Operations Research**, 9(1):77–100, 1982.
- J. F. Bard e J. T. Moore. An algorithm for the discrete bilevel programming problem. **Naval Research Logistics**, 39(3):419–435, apr 1992.

- J. F. Bard, J. Plummer, e J. C. Sourie. A bilevel programming approach to determining tax credits for biofuel production. **European Journal of Operational Research**, 120:30–46, 2000.
- M. S. Bazaraa, H. D. Sherali, e C. M. Shetty. **Linear Complementary Problem, and quadratic, Separable, Fractional, and Geometric Programming**. In: **Nonlinear Programming**. páginas 655–749, John Wiley & Sons, Inc., 2005.
- V. Beresnev. Branch-and-bound algorithm for a competitive facility location problem. **Computers & Operations Research**, 40(8):2062–2070, 2013.
- W. F. Bialas e M. H. Karwan. Two-Level Linear Programming. **Management Science**, 30(8):1004–1020, aug 1984.
- M. A. Boschetti, V. Maniezzo, M. Roffilli, e A. B. Röhrler. Matheuristics: Optimization, Simulation and Control. **Hybrid Metaheuristics**, páginas 171–177, 2009.
- J. Bracken e J. T. McGill. Mathematical programs with optimization problems in the constraints. **Operations Research**, 21:37–44, 1973.
- J. Bracken e J. T. McGill. Defense applications of mathematical programs with optimization problems in the constraints. **Operations Research**, 22:1086–1096, 1974.
- W. Candler e R. Townsley. A linear two-level programming problem. **Computers and Operations Research**, 9(1):59–76, 1982.
- B. Colson, P. Marcotte, e G. Savard. A trust-region method for nonlinear bilevel programming: Algorithm and computational experience. **Computational Optimization and Applications**, 30(3):211–227, 2005.
- B. Colson, P. Marcotte, e G. Savard. An overview of bilevel optimization. **Annals of Operations Research**, 153(1):235–256, 2007.

- I. Constantin e M. Florian. Optimizing frequencies in a transit network: a nonlinear bi-level programming approach. **Intl. Trans. in Operational Research**, 2(2): 149–164, apr 1995.
- N. Costantino, M. Dotoli, M. Falagario, M. P. Fanti, A. M. Mangini, F. Sciancalepore, e W. Ukovich. A hierarchical optimization technique for the strategic design of distribution networks. **Computers and Industrial Engineering**, 66(4):849–864, 2013.
- S. Das e P. N. Suganthan. Differential Evolution: A Survey of the State-of-the-Art. **IEEE Trans. on Evolutionary Computation**, 15(1):4–31, feb 2011.
- K. Deb. An efficient constraint handling method for genetic algorithms. **Computer Methods in Applied Mechanics and Engineering**, 186(2-4):311–338, jun 2000.
- E. D. Dolan e J. J. Moré. Benchmarking optimization software with performance profiles. **Mathematical Programming**, 91(2):201–213, jan 2002.
- P. Hansen, B. Jaumard, e G. Savard. New Branch-and-Bound Rules for Linear Bilevel Programming. **SIAM Journal on Scientific and Statistical Computing**, 13(5):1194–1217, 1992.
- J. J. Júdice e A. M. Faustino. The solution of the linear bilevel programming problem by using the linear complementarity problem. **Investigação Operacional**, 8:77–95, 1988.
- J. J. Júdice e A. M. Faustino. A sequential LCP method for bilevel linear programming. **Annals of Operations Research**, 34(1):89–106, 1992.
- M. Labbé e A. Violin. Bilevel programming and price setting problems. **4OR**, 11(1):1–30, mar 2013.
- F. Legillon, A. Liefooghe, e E. Talbi. CoBRA: A cooperative coevolutionary algo-

- rithm for bi-level optimization. In: **2012 IEEE Congress on Evolutionary Computation**, páginas 1–8. IEEE, jun 2012.
- C. E. Lemke. Bimatrix Equilibrium Points and Mathematical Programming. **Management Science**, 11(7):681–689, may 1965.
- H. Li e Y. Wang. A Hybrid Genetic Algorithm for Solving a Class of Nonlinear Bilevel Programming Problems. In: **Simulated Evolution and Learning**, páginas 408–415. Springer Berlin Heidelberg, 2006.
- H. Li e Y. Wang. A Hybrid Approach for Solving Nonlinear Bilevel Programming Problems Using Genetic Algorithms. In: **2010 Intl. Conference on Computational Intelligence and Security**, páginas 100–103. IEEE, dec 2010.
- H. Li e L. Zhang. A Differential Evolution with Two Mutation Strategies and a Selection Based on an Improved Constraint-Handling Technique for Bilevel Programming Problems. **Mathematical Problems in Engineering**, 2014, 2014.
- M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, e M. Birattari. The irace package, iterated race for automatic algorithm configuration. Relatório Técnico TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.
- A. Massaro e M. Pelillo. **A Pivoting-Based Heuristic for the Maximum Clique Problem**. In: N. Hadjisavvas e P. M. Pardalos (editores), **Advances in Convex Analysis and Global Optimization: Honoring the Memory of C. Caratheodory (1873–1950)**. páginas 383–394, Springer US, Boston, MA, 2001.
- K. Moshirvaziri, M. A. Amouzegar, e S. E. Jacobsen. Test problem construction for linear bilevel programming problems. **Journal of Global Optimization**, 8(3):235–243, 1996.
- K. G. Murty. **Linear Complementarity, Linear and Nonlinear Programming**, volume 38. Heldermann Verlag, 1997.

- K. G. Murty e S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. **Mathematical Programming**, 39(2):117–129, 1987. ISSN 14364646.
- V Oduguwa e R Roy. Bi-Level Optimisation Using Genetic Algorithm. In: **Proc. of the 2002 IEEE Intl. Conf. on Artificial Intelligence Systems (ICAIS'02)**, páginas 322–327, Washington, DC, USA, 2002. IEEE Computer Society.
- J. V. Outrata. On the numerical solution of a class of stackelberg problems. **Zeitschrift für Operations Research**, 34(4):255–277, 1990.
- J. Rajesh, K. Gupta, H. S. Kusumakar, V. K. Jayaraman, e B. D. Kulkarni. A tabu search based approach for solving a class of bilevel programming problems in chemical engineering. **Journal of Heuristics**, 9:307–319, 2003.
- G. Savard e J. Gauvin. The steepest descent direction for the nonlinear bilevel programming problem. **Operations Research Letters**, 15(5):265–272, 1994.
- K. Shimizu e E. Aiyoshi. A new computational method for stackelberg and min-max problems by use of a penalty method. **IEEE Trans. on Automatic Control**, AC-26(2):460–466, 1981.
- A. Sinha, P. Malo, e K. Deb. Unconstrained scalable test problems for single-objective bilevel optimization. In: **2012 IEEE Congress on Evolutionary Computation (CEC)**, páginas 1–8, 2012.
- A. Sinha, P. Malo, e K. Deb. Test Problem Construction for Single-Objective Bilevel Optimization. **Evolutionary Computation**, páginas 1–35, 2013.
- H. Von Stackelberg. **The Theory of the Market Economy**. Oxford University Press, 1952. Traduzido para o inglês de **Marktform und Gleichgewicht**. Springer-Verlag, Berlin, 1934.

- R. M. Storn e K. V. Price. Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Relatório Técnico TR-95-012, ICSI, 1995.
- E. Talbi. **Metaheuristics: From Design to Implementation**. Wiley Publishing, 2009.
- E. Talbi. **Metaheuristics for Bi-level Optimization**, volume 482 of **Studies in Computational Intelligence**. Springer Berlin Heidelberg, 2013.
- H. Tuy, A. Migdalas, e N. T. Hoai-Phuong. A novel approach to bilevel nonlinear programming. **Journal of Global Optimization**, 38(4):527–554, 2007.
- L. F. C. N. Vicente. **Programação de Dois Níveis**. Tese de Doutorado, Universidade de Coimbra, 1992.
- Y. Wang, Y.-C. Jiao, e H. Li. An Evolutionary Algorithm for Solving Nonlinear Bilevel Programming Based on a New Constraint-Handling Scheme. **IEEE Trans. on Systems, Man and Cybernetics, Part C (Applications and Reviews)**, 35(2):221–232, 2005.
- U.-P. Wen e Y.-H. Yang. Algorithms for solving the mixed integer two-level linear programming problem. **Computers & Operations Research**, 17(2):133–142, jan 1990.
- D. J. White e G. Anandalingam. A penalty function approach for solving bi-level linear programs. **Journal of Global Optimization**, 3(4):397–419, 1993.