



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS FÍSICAS E MATEMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA PURA E APLICADA

Rafaela Filippozzi

**Métodos para o problema de inclusão no envoltório convexo**

Florianópolis  
2023

Rafaela Filippozzi

**Métodos para o problema de inclusão no envoltório convexo**

Tese submetida ao Programa de Pós-Graduação em Matemática Pura e Aplicada da Universidade Federal de Santa Catarina para a obtenção do título de doutor em matemática pura e aplicada..

Orientador: Prof. Douglas Soares Gonçalves, Dr.

Coorientador: Prof. Luiz Rafael dos Santos, Dr.

Florianópolis

2023

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Filippozzi, Rafaela

Métodos para o problema de inclusão no envoltório convexo / Rafaela Filippozzi ; orientador, Douglas Soares Gonçalves, coorientador, Luiz Rafael dos Santos, 2023.

86 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro de Ciências Físicas e Matemáticas, Programa de Pós-Graduação em Matemática Pura e Aplicada, Florianópolis, 2023.

Inclui referências.

1. Matemática Pura e Aplicada. 2. Otimização Convexa. 3. Métodos do tipo Frank-Wolfe. 4. Algoritmo do Triângulo. 5. Problema de inclusão no envoltório convexo. I. Soares Gonçalves, Douglas . II. dos Santos, Luiz Rafael . III. Universidade Federal de Santa Catarina. Programa de Pós Graduação em Matemática Pura e Aplicada. IV. Título.

Rafaela Filippozzi

**Métodos para o problema de inclusão no envoltório convexo**

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Max Leandro Nobre Gonçalves, Dr.  
Instituição Universidade Federal de Goiás

Prof. Luis Felipe da Rocha Bueno, Dr.  
Instituição Universidade Federal de São Paulo

Prof. Juliano de Bem Francisco, Dr.  
Instituição Universidade Federal de Santa Catarina

Prof. Maicon Marques Alves, Dr.  
Instituição Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de doutor em matemática pura e aplicada..

---

Coordenação do Programa de  
Pós-Graduação

---

Prof. Douglas Soares Gonçalves, Dr.  
Orientador

Florianópolis, 2023.

Este trabalho é dedicado a (ao) minha (meu) filha (filho)  
Sol ao qual não pude ver mas pude sentir iluminar.

## AGRADECIMENTOS

Gostaria de iniciar agradecendo pelo amadurecimento intelectual e emocional que adquiri durante os quatro anos de meu doutorado. Quando cheguei pela primeira vez na Universidade Federal de Santa Catarina (UFSC), para um curso de verão em 2016, como estudante de graduação, eu não tinha ideia da jornada acadêmica que estava por vir.

Me sinto honrada por ter como coorientador um professor que me conheceu no início dessa jornada, e que até mesmo realizou minha matrícula na graduação. Agradeço imensamente ao professor Luiz Rafael dos Santos por me inspirar a sempre ir além e acreditar em meu potencial.

Além de ter um excepcional coorientador, tenho um grande mestre como orientador, o professor Douglas Soares Gonçalves. Quero expressar minha gratidão ao professor Douglas por confiar em mim, mesmo durante a pandemia e nos momentos em que a síndrome do impostor ameaçava desencorajar-me de prosseguir com meu doutorado. Sua inspiração diária e excelência como orientador e professor são inestimáveis.

Sou muito grata ao apoio financeiro fornecido pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) para o desenvolvimento de minha tese. Gostaria de agradecer a todos os professores do Departamento de Matemática da UFSC, em especial aos que ministraram minhas aulas e seminários, bem como ao professor Antônio Carlos Gardel Leitão, que me proporcionou a oportunidade de expandir meus horizontes em áreas de pesquisa distintas e inclusive publicar artigos em uma área que agora ocupa um lugar especial em meu coração.

Gostaria de expressar minha profunda gratidão aos meus familiares e amigos nesta seção de agradecimentos. Sinto-me imensamente grata por ter pessoas maravilhosas ao meu redor que me apoiaram durante todo o processo do doutorado. Sem o incentivo e suporte deles, não teria sido possível participar de um congresso internacional e apresentar parte deste trabalho, que foi uma das melhores experiências da minha vida.

Gostaria de agradecer especialmente ao meu companheiro, Rafael Herger Pinto, pelas palavras de incentivo e apoio constantes ao longo do caminho. Sua presença e encorajamento foram fundamentais para eu chegar até aqui. Estou muito agradecida por tê-lo em minha vida.

Por fim, agradeço à minha espiritualidade por me fazer acreditar mais em mim mesma, tornando-me agora uma MULHER DOUTORA EM MATEMÁTICA, empoderada e confiante.

*“A vida é cheia de desafios. E todo esforço tem uma recompensa. E toda recompensa precisa de esforços.  
Onde os esforços baseiam-se em querer, em tentar entender, em enfrentar e persistir até chegar onde  
sempre sonhou estar.”  
(Pri Ferraz, 2021)*

## RESUMO

O problema de inclusão no envoltório convexo consiste em determinar se um ponto pode ser escrito como combinação convexa de um conjunto finito de pontos dados, isto é, se tal ponto pertence ao envoltório convexo dos dados pontos. Este é um problema de decisão com importantes aplicações em geometria computacional e em fundamentos de programação linear. Neste trabalho, propomos o uso de métodos de otimização de primeira ordem para resolver o problema. Mostramos que métodos do tipo Frank-Wolfe e gradiente projetado, quando equipados com novos critérios de parada, são eficazes para tal problema de decisão, e apresentam desempenho favorável em relação a um algoritmo recente específico ao problema, chamado *Algoritmo do Triângulo*. Discutimos as conexões entre este algoritmo e Frank-Wolfe, mostrando que o primeiro pode ser interpretado como um Frank-Wolfe inexato. Apesar dessa semelhança, o Algoritmo do Triângulo é fortemente baseado em um teorema de alternativas conhecido como dualidade de distâncias. Usando tal teorema, desenvolvemos novos critérios de parada para métodos do tipo Frank-Wolfe e gradiente projetado, especializando-os para o problema de inclusão. Reportamos experimentos numéricos executados em exemplares artificiais, cuidadosamente gerados para cobrir diferentes cenários, que indicam qual algoritmo é preferível de acordo com a geometria do envoltório convexo e a posição relativa do ponto a ser consultado. No que diz respeito a aplicações potenciais, apresentamos dois exemplos ilustrativos, um relacionado a problemas de viabilidade de programação linear e outro relacionado a problemas de classificação de imagem. Além disso, também consideramos uma reformulação do problema na qual o ponto a ser consultado é a origem e os pontos do conjunto dado têm norma um. Investigamos uma propriedade que, quando satisfeita a cada iteração, permite melhorar a complexidade de iteração do Algoritmo do Triângulo. Introduzimos uma *heurística* que busca favorecer a ocorrência de tal propriedade e resultados numéricos preliminares atestam sua efetividade.

**Palavras-chave:** Otimização Convexa; Métodos do tipo Frank-Wolfe; Algoritmo do Triângulo; Problema de inclusão no envoltório convexo.



## ABSTRACT

The convex hull membership problem consists in deciding whether a point can be written as a convex combination of a finite set of given points, that is, whether such a point belongs to the convex hull of the given points. This is a decision problem with important applications in computational geometry and in foundations of linear programming. In this work we propose the use of first-order optimization methods to solve this problem. We show that, Frank-Wolfe type methods and Projected Gradient methods, when equipped with new stopping criteria, are effective for such a decision problem and perform favorably against a recent problem-specific algorithm called Triangle Algorithm. We discuss the connections between this algorithm and Frank-Wolfe, showing that the former can be interpreted as an inexact Frank-Wolfe. Despite this similarity, the Triangle Algorithm is strongly based on a theorem of alternatives known as distance duality. By using this theorem, we devise new stopping criteria integrated for Frank-Wolfe type and Projected Gradient methods, specializing them to the membership decision problem. We report numerical experiments on artificial instances, carefully designed to cover different scenarios, that indicate which algorithm is preferable according to the geometry of the convex hull and the relative position of the query point. Concerning potential applications, we present two illustrative examples, one related to linear programming feasibility problems and another related to image classification problems. Furthermore, we also consider a reformulation of the problem in which the query point is the origin and the points of the given set have norm one. We investigated a property that, when satisfied at each iteration, allows us to improve the iteration complexity of the Triangle Algorithm. We introduce a heuristic that seeks to favor the occurrence of such a property and preliminary numerical results corroborates its effectiveness.

**Keywords:** Convex Optimization; Frank-Wolfe-type methods; Triangle Algorithm; Convex hull membership problem.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>2</b>	<b>PRELIMINARES</b>	<b>14</b>
2.1	NOÇÕES DE CONVEXIDADE	14
2.2	NOÇÕES DE OTIMIZAÇÃO NÃO-LINEAR	16
2.3	OTIMIZAÇÃO SOBRE CONJUNTOS CONVEXOS	18
<b>2.3.1</b>	<b>Métodos do tipo Frank-Wolfe</b>	<b>19</b>
<b>2.3.2</b>	<b>Métodos do tipo Gradiente Projetado</b>	<b>25</b>
<b>3</b>	<b>MÉTODOS DE PRIMEIRA ORDEM PARA O PROBLEMA DE INCLUSÃO NO ENVOLTÓRIO CONVEXO</b>	<b>28</b>
3.1	O PROBLEMA DE INCLUSÃO NO ENVOLTÓRIO CONVEXO	28
3.2	ALGORITMO DO TRIÂNGULO	29
3.3	FRANK-WOLFE APLICADO AO CHMP	39
<b>3.3.1</b>	<b>Frank-Wolfe como um Algoritmo do Triângulo Ganancioso</b>	<b>41</b>
3.4	GRADIENTE PROJETADO APLICADO AO CHMP	43
3.5	CRITÉRIOS DE PARADA ESPECÍFICOS PARA CHMP	43
3.6	EXPERIMENTOS NUMÉRICOS	45
<b>3.6.1</b>	<b>Exemplos Artificiais do CHMP</b>	<b>46</b>
<b>3.6.2</b>	<b>Problema de Factibilidade de Programação Linear</b>	<b>51</b>
<b>3.6.3</b>	<b>Um exemplo de CHMP em classificação de imagens</b>	<b>54</b>
3.7	CONCLUSÕES DO CAPÍTULO	56
<b>4</b>	<b>CHMP ESFÉRICO E UMA VARIANTE <math>\mathcal{O}(1/\varepsilon)</math> DO ALGORITMO DO TRIÂNGULO</b>	<b>57</b>
4.1	CHMP ESFÉRICO	57
4.2	UMA NOVA HEURÍSTICA PARA O CHMP-ESFÉRICO	63
4.3	RESOLVENDO O SUBPROBLEMA	67
4.4	EXPERIMENTOS NUMÉRICOS PARA O TA-ESFÉRICO	69
<b>4.4.1</b>	<b>Investigação empírica da complexidade de iteração</b>	<b>69</b>
<b>4.4.2</b>	<b>Exemplos artificiais para o CHMP Esférico e tempo de execução</b>	<b>70</b>
4.5	CONCLUSÕES DO CAPÍTULO	71
<b>5</b>	<b>CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS</b>	<b>73</b>
	<b>REFERÊNCIAS</b>	<b>75</b>
	<b>APÊNDICE A – DEMONSTRAÇÃO LEMA 3.2</b>	<b>81</b>
	<b>APÊNDICE B – DEDUÇÃO DAS CONSTANTES DA COMPLEXIDADE DE ITERAÇÃO (TABELA 1)</b>	<b>82</b>
	<b>APÊNDICE C – TABELAS ADICIONAIS DA SEÇÃO 3.6.1</b>	<b>84</b>

## 1 INTRODUÇÃO

Seja  $\mathcal{A} := \{v_1, v_2, \dots, v_n\} \subset \mathbb{R}^m$  e considere um ponto  $p \in \mathbb{R}^m$ . O *problema de inclusão no envoltório convexo* (ou CHMP, do inglês *convex hull membership problem*) consiste em decidir se  $p \in \text{conv}(\mathcal{A})$ , em que  $\text{conv}(\mathcal{A})$  denota o envoltório convexo de  $\mathcal{A}$ . Este problema está relacionado a conceitos fundamentais em programação linear e encontra aplicações importantes em geometria computacional [13, 37, 53].

Na literatura há métodos específicos para resolver o CHMP, dentre os quais podemos destacar Quick Hull [3], Graham's Scan [28] e um recente algoritmo proposto por Kalantari [37] inspirado em ideias geométricas, ao qual chamaremos de *Algoritmo do Triângulo* (ou TA, do inglês *Triangle Algorithm*).

Diferente dos dois primeiros, que normalmente são usados em dimensão 2 ou 3, o Algoritmo do Triângulo pode ser aplicado para problemas em qualquer dimensão finita. Em termos de complexidade de iteração, Kalantari [37, Theorem 9] mostrou que o TA encontra um ponto  $p_\varepsilon \in \text{conv}(\mathcal{A})$  a distância no máximo  $\varepsilon$  de  $p$ , ou determina que  $p \notin \text{conv}(\mathcal{A})$ , em  $\mathcal{O}(1/\varepsilon^2)$  iterações. Este algoritmo também foi empregado com sucesso para resolver *outros problemas importantes relacionados ao CHMP*. Por exemplo, [39] considerou o problema mais geral de separação de dois conjuntos convexos compactos (CHMP é um caso especial, quando um dos conjuntos é um único ponto). Mais ainda, em [2] foi mostrado que aplicando o TA diversas vezes, é possível enumerar todos os pontos extremos do envoltório convexo, mesmo que em dimensões grandes. Para outros trabalhos recentes envolvendo o Algoritmo do Triângulo, veja [38, 40, 41, 43].

É interessante mencionar que o Algoritmo do Triângulo explora um teorema de alternativas, chamado de *dualidade de distâncias*:

- ou bem para todo  $p' \in \text{conv}(\mathcal{A})$ , existe  $v_j \in \mathcal{A}$  tal que  $d(v_j, p) \leq d(v_j, p')$ ;
- ou então existe  $p' \in \text{conv}(\mathcal{A})$  tal que o hiperplano bissetor ortogonal ao segmento que une  $p'$  a  $p$  separa  $p$  de  $\text{conv}(\mathcal{A})$ .

Por outro lado, também é possível abordar o CHMP através de modelos de programação linear e quadrática. Definindo  $A := [v_1 \ v_2 \ \dots \ v_n] \in \mathbb{R}^{m \times n}$  como a matriz na qual cada coluna é um dos  $n$  pontos de  $\mathcal{A}$ , podemos ver que  $p \in \text{conv}(\mathcal{A})$  se e somente se  $p$  é uma combinação convexa das colunas de  $A$ . Seja  $e \in \mathbb{R}^n$  o vetor com todas as componentes iguais a um. Podemos descrever formalmente o problema de inclusão no envoltório convexo como o seguinte *problema de decisão*:

$$\text{Existe algum } x \in \mathbb{R}^n \text{ tal que } Ax = p, e^T x = 1, x \geq 0? \quad (1)$$

O problema (1) é um problema de factibilidade de programação linear cuja resposta afirmativa garante  $p \in \text{conv}(\mathcal{A})$ . Tal problema de factibilidade também pode ser resolvido pelo seguinte problema de programação quadrática

$$\min_{x \in \Delta_n} \frac{1}{2} \|Ax - p\|^2, \quad (2)$$

em que  $\Delta_n := \{x \in \mathbb{R}^n \mid e^T x = 1, x \geq 0\}$  é o simplex unitário em  $\mathbb{R}^n$ . Claramente,  $p \in \text{conv}(\mathcal{A})$  quando o valor ótimo de (2) é zero. Além disso, se  $x$  é um ponto viável de (2), então  $y := Ax \in \text{conv}(\mathcal{A})$ . Portanto, outra possível formulação é dada por

$$\begin{aligned} \min_{y \in \mathbb{R}^m} \quad & \frac{1}{2} \|y - p\|^2 \\ \text{s. a} \quad & y \in \text{conv}(\mathcal{A}), \end{aligned} \tag{3}$$

que atinge um valor ótimo zero se, e somente se,  $p \in \text{conv}(\mathcal{A})$ .

As formulações (1), (2) e (3) mostram que podemos resolver o CHMP com métodos *clássicos de programação linear ou quadrática*. Com a formulação (2), por exemplo, podemos utilizar métodos do tipo gradiente projetado [29, 30, 48, 56], uma vez que a projeção no simplex unitário pode ser calculada em  $\mathcal{O}(n \log n)$  [16], ou seja, em tempo computacional razoável.

Já para resolver o problema (3), métodos do tipo Frank-Wolfe (FW) [15, 23, 35, 45] (conhecidos também como métodos de gradiente condicional), se mostram mais adequados, podendo ser aplicados com custo por iteração de  $\mathcal{O}(mn)$ .

Tendo em perspectiva estes últimos métodos, os quais fazem uso de informações de derivadas de primeira ordem, é que esta tese é desenvolvida, utilizando as características geométricas do CHMP e do TA a fim de torna-los mais eficazes para o problema de inclusão no envoltório convexo. As contribuições que trazemos são resumidas a seguir.

Na primeira contribuição deste trabalho investigamos como métodos de primeira ordem, como os métodos de Frank-Wolfe e Gradiente Projetado podem ser empregados para resolução eficiente do CHMP. Mais especificamente:

- (i) Estudamos as semelhanças entre Frank-Wolfe e o TA: trabalhos anteriores sobre o Algoritmo do Triângulo [37, 39] mencionam brevemente o uso do método de FW para CHMP e contrastam seu mecanismo de iteração com o do primeiro. Mostramos que FW com um critério de parada baseado na dualidade de distâncias (útil no caso  $p \notin \text{conv}(\mathcal{A})$ ) pode ser interpretado como uma versão gananciosa do TA: tal versão é equivalente a um algoritmo atribuído a von Neumann, como comunicado por [18]. Mais ainda, mostramos que o Algoritmo do Triângulo é um método tipo Frank-Wolfe, com resolução inexata dos subproblemas.
- (ii) Com base nessa relação, propomos novas variantes do TA, bem como variantes de Frank-Wolfe que aproveitam a dualidade de distâncias.
- (iii) Propomos utilizar da dualidade de distâncias para criar *critérios de parada* adequados para métodos tipo Frank-Wolfe e outros métodos de primeira ordem, como o gradiente projetado, a fim de torna-los eficientes para a resolução do CHMP.

- (iv) Comparamos o desempenho das novas variantes com suas versões clássicas e outros métodos de primeira ordem, como métodos de gradiente projetado espectral [8, 9], por meio de experimentos numéricos cuidadosamente planejados, os quais cobrem não apenas cenários favoráveis, mas também *adversários* para os algoritmos, indicando em qual situação um algoritmo é preferível a outro. Salientamos que trabalhos anteriores [2, 47] relatam experimentos numéricos comparando o TA com Frank-Wolfe, porém critérios de parada especializados para CHMP não foram usados para FW. Visando uma comparação justa, integramos o FW (e os demais métodos de primeira ordem) com critérios de parada especializados. Além disso, até onde sabemos, nenhum desses trabalhos considerou os métodos Away-Step Frank-Wolfe (ASFW) [35, 45] e Gradiente Projetado nos experimentos computacionais.
- (v) Apresentamos duas potenciais aplicações que envolvem a resolução de um CHMP: problemas de factibilidade em programação linear e problemas de classificação de imagens.

Estes resultados foram publicados recentemente no periódico *European Journal of Operational Research*, pela autora desta tese e seus orientadores [22].

Na segunda contribuição deste trabalho, a qual esta em fase de preparação para ser submetido para publicação, consideramos um problema equivalente ao CHMP, conhecido como CHMP esférico. Seja  $\mathcal{S} := \{s_1, \dots, s_n\} \subset \mathbb{R}^m$  tal que  $\|s_i\| = 1$ , para todo  $i = 1, \dots, n$ . O CHMP esférico consiste em determinar se a origem, isto é,  $p = 0 \in \mathbb{R}^m$  pertence a  $\text{conv}(\mathcal{S})$ .

Em [42], foi provada a equivalência entre CHMP e CHMP esférico e proposto um TA específico para o segundo. Assim como a versão clássica, tal algoritmo também tem a mesma complexidade de iteração de  $\mathcal{O}(1/\varepsilon^2)$  para encontrar uma solução aproximada para o problema (i.e., encontrar  $p_k \in \text{conv}(\mathcal{S})$  tal que  $\|p_k\| < \varepsilon$  ou determinar que  $0 \notin \text{conv}(\mathcal{S})$ ). Porém, sob hipóteses adicionais, em particular se cada iteração do TA satisfizer uma certa propriedade (veja Capítulo 4), é possível melhorar a complexidade de iteração para  $\mathcal{O}(1/\varepsilon)$ .

Buscamos então desenvolver um procedimento heurístico, a ser integrado no Algoritmo do Triângulo, capaz de favorecer a ocorrência de tal propriedade a cada iteração. Em experimentos numéricos preliminares tal heurística mostrou-se efetiva. Em verdade, em todos os testes realizados, sempre que acionada, a heurística retornou um novo iterado cumprindo a propriedade desejada. Deixamos assim a correteza do procedimento proposto como uma conjectura.

Esta tese está organizada da seguinte forma. No Capítulo 2 apresentamos notações e conceitos preliminares sobre convexidade e otimização contínua que serão úteis no decorrer do texto. O Capítulo 3 traz a primeira parte da contribuição deste trabalho, mostrando como métodos de otimização de primeira ordem podem ser eficientes para o problema de decisão CHMP. No Capítulo 4, apresentamos uma formulação equivalente para o problema

---

de inclusão no envoltório convexo, proposta por Kalantari e Zhang [42], assim como uma variante do algoritmo do Triângulo para tal caso. Além disso, propomos uma heurística para acelerar essa variante e exibimos experimentos numéricos para validar esta nova proposta. Conclusões e perspectivas futuras são encontradas no Capítulo 5.

## 2 PRELIMINARES

Neste capítulo serão apresentados alguns conceitos, resultados teóricos e algoritmos que são utilizados ao longo deste trabalho. Tal conteúdo é sobretudo baseado nas referências [5, 7, 33, 52]. Quanto a notações, denotamos por  $e \in \mathbb{R}^n$  o vetor com todas as componentes iguais a um, por  $v^T p$  o produto interno Euclidiano entre os vetores  $v, p \in \mathbb{R}^m$  e por  $\|\cdot\|$  sua norma induzida. Também denotamos a distância Euclidiana entre  $v$  e  $p$  como  $d(v, p) := \|v - p\|$ , a bola centrada em  $v$  com raio  $\rho$  como  $B_\rho(v) := \{p \in \mathbb{R}^m \mid d(v, p) < \rho\}$  e a combinação convexa ou segmento entre  $v$  e  $p$  como  $[v, p]$ . Para indicar a ordem de complexidade, seja de convergência ou de iteração, utilizamos  $\mathcal{O}$ .

### 2.1 NOÇÕES DE CONVEXIDADE

Nessa seção trazemos algumas definições e resultados sobre convexidade de forma extremamente sucinta, o suficiente para o desenvolvimento de resultados posteriores deste trabalho. Para um tratamento didático, mais detalhado e com exemplos, veja [7, 33, 52].

**Definição 2.1.** Um subconjunto não-vazio  $C \subset \mathbb{R}^m$  é dito um *conjunto convexo* quando para quaisquer  $x, y \in C$  tem-se que

$$\alpha x + (1 - \alpha)y \in C, \quad \forall \alpha \in [0, 1].$$

**Definição 2.2.** Sejam  $v_1, v_2, \dots, v_n$  e  $v$  vetores de  $\mathbb{R}^m$ . Quando  $v = \sum_{i=1}^n \alpha_i v_i$ , com  $\alpha_i \geq 0$ , para  $i = 1, \dots, n$ , dizemos que  $v$  é *combinação cônica* dos vetores  $v_1, v_2, \dots, v_n$ .

**Definição 2.3.** Sejam  $v_1, v_2, \dots, v_n$  e  $v$  vetores de  $\mathbb{R}^m$ . Quando  $v = \sum_{i=1}^n \alpha_i v_i$ , com  $\alpha_i \geq 0$ , para  $i = 1, \dots, n$ , e  $e^T \alpha = 1$ , dizemos que  $v$  é *combinação convexa* dos vetores  $v_1, v_2, \dots, v_n$ .

**Definição 2.4.** Seja  $\mathcal{A} = \{v_1, v_2, \dots, v_n\}$  um conjunto finito de pontos. O conjunto de todas as combinações convexas de qualquer número de elementos de  $\mathcal{A}$  é denominado *envoltório convexo*, *casca convexa* ou *fecho convexo* do conjunto  $\mathcal{A}$ , o qual será denotado por  $\text{conv}(\mathcal{A})$ .

É possível mostrar que  $\text{conv}(\mathcal{A})$  é o menor conjunto convexo que contém  $\mathcal{A}$  (como ilustrado na Figura 1), ou equivalentemente, a intersecção de todos os conjuntos convexas que contêm  $\mathcal{A}$ . Para mais detalhes, veja [7, Appendix B].

**Definição 2.5.** Seja  $C$  um conjunto convexo. Dizemos que  $v$  é um *ponto extremo* de  $C$ , se  $v$  não pode ser escrito como combinação convexa de outros elementos de  $C$ .

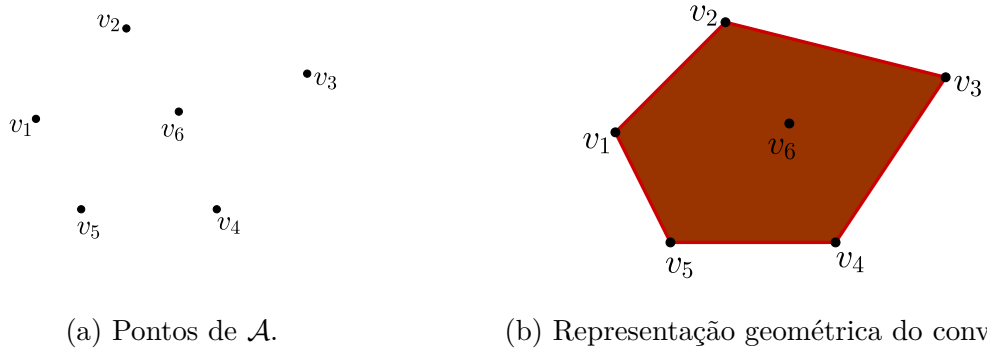


Figura 1 – Ilustração de Envoltório Convexo.

**Definição 2.6.** Seja  $C \subset \mathbb{R}^m$  um conjunto convexo não-vazio. Uma função  $f : C \rightarrow \mathbb{R}$  é dita *convexa* se para quaisquer  $x, y \in C$ :

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \quad \forall \alpha \in [0, 1].$$

Na inequação acima, se valer a desigualdade estrita, para todo  $x \neq y$  e  $\alpha \in (0, 1)$ , então dizemos que  $f$  é *estritamente convexa*.

**Proposição 2.7** ([7, Proposition B.3]). *Sejam  $C \subset \mathbb{R}^m$  um conjunto convexo não-vazio e  $f : \mathbb{R}^m \rightarrow \mathbb{R}$  uma função diferenciável em  $\mathbb{R}^m$ . A função  $f$  é convexa sobre  $C$  se, e somente se,*

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) \quad \forall x, y \in C.$$

**Definição 2.8.** Seja  $C \subset \mathbb{R}^m$  um conjunto convexo não-vazio. Uma função  $f : C \rightarrow \mathbb{R}$ , diferenciável em  $\mathbb{R}^m$ , é dita *fortemente convexa* se existe  $\mu > 0$  tal que, para quaisquer  $x, y \in C$ :

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{\mu}{2}\|y - x\|^2.$$

**Teorema 2.9** (Teorema da Projeção [7, Proposition 2.1.3]). *Seja  $C \subset \mathbb{R}^m$  um conjunto não-vazio, convexo e fechado.*

(a) *Para todo  $x \in \mathbb{R}^m$ , existe um único elemento  $P_C(x) \in C$  (chamado de projeção de  $x$  em  $C$ ) tal que  $\|P_C(x) - x\| \leq \|z - x\|, \forall z \in C$ .*

(b) *Dado qualquer  $x \in \mathbb{R}^m$ ,  $z = P_C(x)$  se, e somente se,*

$$(y - z)^T(x - z) \leq 0, \quad \forall y \in C.$$

(c) *A aplicação  $P_C(\cdot)$  é contínua e não expansiva, isto é*

$$\|P_C(x) - P_C(y)\| \leq \|x - y\|, \quad \forall x, y \in \mathbb{R}^m.$$

(d) *Dado  $x \in \mathbb{R}^m$ , para todo  $y$  em  $C$  temos que  $\|y - P_C(x)\| \leq \|y - x\|$ .*



**Definição 2.10.** Um *hiperplano* em  $\mathbb{R}^m$  é um conjunto da forma  $\{x \in \mathbb{R}^m \mid a^T x = b\}$ , em que  $a \in \mathbb{R}^m$ ,  $a \neq 0$ , e  $b \in \mathbb{R}$ . Os conjuntos  $\{x \in \mathbb{R}^m \mid a^T x \geq b\}$ ,  $\{x \in \mathbb{R}^m \mid a^T x \leq b\}$ , são chamados de *semi-espacos* associados ao hiperplano.

**Definição 2.11.** Um subconjunto de  $\mathbb{R}^m$  que pode ser expresso como a interseção de um número finito de semi-espacos é chamado de *poliedro*.

**Teorema 2.12** (Teorema do Hiperplano Separador [7, Proposition B.13]). *Seja  $C \subset \mathbb{R}^m$  um conjunto não-vazio, convexo e fechado e  $x \notin C$ . Então existe um vetor  $a \in \mathbb{R}^m \setminus \{0\}$  tal que*

$$a^T y > a^T x, \quad \forall y \in C.$$

O Teorema 2.12 nos diz que dado um conjunto convexo fechado  $C$  e um ponto  $x \notin C$ , existe um hiperplano que separa estritamente  $x$  do conjunto  $C$ . Não é difícil mostrar que o hiperplano  $\{y \in \mathbb{R}^m \mid a^T y = \beta\}$ , em que  $a := P_C(x) - x$  e  $\beta := (P_C(x) - x)^T P_C(x)$ , além de separar  $x$  de  $C$ , tangencia  $C$  em  $P_C(x)$ : a este hiperplano damos o nome de *hiperplano suporte*.

## 2.2 NOÇÕES DE OTIMIZAÇÃO NÃO-LINEAR

A seguir apresentamos definições e resultados fundamentais em otimização contínua não linear suave. Novamente tais conceitos serão apresentados de maneira bastante resumida; para mais detalhes recomendamos [7, 48].

Chamamos de problema geral de otimização não-linear:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s. a} \quad & h(x) = 0, \\ & g(x) \leq 0, \end{aligned} \tag{5}$$

em que  $f : \mathbb{R}^m \rightarrow \mathbb{R}$ ,  $h : \mathbb{R}^m \rightarrow \mathbb{R}^n$ ,  $g : \mathbb{R}^m \rightarrow \mathbb{R}^p$ , são funções continuamente diferenciáveis. Seja  $C = \{x \in \mathbb{R}^m \mid h(x) = 0, g(x) \leq 0\}$ . Qualquer ponto  $x \in C$  é chamado de ponto factível (ou viável) de (5).

**Definição 2.13.** Um ponto  $x \in C$  é dito *minimizador local* de  $f$  em  $C$ , se existe  $\varepsilon > 0$  tal que  $f(x) \leq f(y)$  para todo  $y \in C$  satisfazendo  $\|x - y\| \leq \varepsilon$ . Um vetor  $x \in C$  é chamado de *minimizador global* de  $f$  em  $C$  se  $f(x) \leq f(y)$  para todo  $y \in C$ .

**Definição 2.14.** Seja  $\bar{x}$  um ponto factível para (5). Uma restrição  $g_j(x) \leq 0$ , para  $j = 1, \dots, p$  é dita *restrição ativa* em  $\bar{x}$  quando  $g_j(\bar{x}) = 0$ . Se  $g_j(\bar{x}) < 0$ , dizemos que a restrição é inativa.

Denotamos por  $I(\bar{x}) \subset \{1, \dots, p\}$  o conjunto de índices das restrições de desigualdade ativas em  $\bar{x}$ , i.e.,  $I(\bar{x}) = \{j \in \{1, \dots, p\} \mid g_j(\bar{x}) = 0\}$ .

**Definição 2.15.** Considere o problema (5). Dizemos que o ponto factível  $\bar{x}$  é *regular* se  $\{\nabla h_i(\bar{x})\}_{i=1}^n \cup \{\nabla g_j(\bar{x})\}_{j \in I(\bar{x})}$  é um conjunto linearmente independente.

Para o problema geral de otimização não-linear, minimizadores locais que satisfazem certas condições de regularidade devem cumprir as conhecidas condições de Karush-Kuhn-Tucker (KKT), as quais iremos apenas enunciar abaixo.

**Teorema 2.16** (Condições KKT [7, Proposition 3.3.1]). *Seja  $\bar{x}$  minimizador local de (5). Se  $\bar{x}$  é regular, então existem  $\lambda \in \mathbb{R}^n$ ,  $\mu \in \mathbb{R}_+^p$  tais que*

$$\begin{cases} \nabla f(\bar{x}) + \sum_{i=1}^n \lambda_i \nabla h_i(\bar{x}) + \sum_{j=1}^p \mu_j \nabla g_j(\bar{x}) = 0 \\ \mu_j g_j(\bar{x}) = 0, \quad j = 1, \dots, p \\ h(\bar{x}) = 0, \quad g(\bar{x}) \leq 0. \end{cases}$$

Uma discussão detalhada sobre condições KKT, outras condições de regularidade e demonstrações do resultado acima podem ser encontradas em [7, 48, 52].

A definição a seguir é de extrema importância para motivação dos algoritmos que serão apresentados na próxima seção e para alguns resultados teóricos que veremos a seguir.

**Definição 2.17.** Considere o problema (5) e  $x \in C$ . Dizemos que  $d \in \mathbb{R}^m$  é uma direção factível a partir de  $x$ , se existe  $\bar{\gamma} > 0$  tal que  $x + \gamma d \in C$ ,  $\forall \gamma \in [0, \bar{\gamma})$ . Dizemos que  $d$  é uma direção de descida para  $f$  a partir de  $x$  se existe  $\varepsilon > 0$  tal que  $f(x + \gamma d) < f(x)$ ,  $\forall \gamma \in (0, \varepsilon]$ .

Sendo assim, se  $x^* \in C$  é minimizador local de (5), então não existe direção  $d \in \mathbb{R}^m$  factível e de descida. Uma condição suficiente para que  $d_k$  seja uma direção de descida é a do próximo resultado.

**Proposição 2.18** ([24, Proposição 4.1]). *Seja  $f : \mathbb{R}^m \rightarrow \mathbb{R}$  uma função continuamente diferenciável. Se  $\nabla f(x_k)^T d_k < 0$ , então  $d_k$  é direção de descida para  $f$  a partir de  $x_k$ .*

Considere o problema (5), com  $x_k \in C$  e  $d_k$  direção factível e de descida. Métodos de busca linear em direções factíveis (e de descida) determinam o iterado seguinte como  $x_{k+1} = x_k + \gamma_k d_k$ , fazendo uma escolha adequada do tamanho de passo  $\gamma_k$ . Há várias estratégias para escolher o tamanho de passo  $\gamma_k$  satisfazendo  $f(x_k + \gamma_k d_k) < f(x_k)$ , entre eles: passo fixo, busca linear exata, busca de Armijo, *backtracking* e estratégias de interpolação [7, 34]. A seguir apresentamos sucintamente duas maneiras que são utilizadas mais adiante nesta tese.

A primeira maneira é escolher um tamanho de passo que diminui a cada iteração  $k$ , ou seja, que converge para 0, isto é,  $\gamma_k \rightarrow 0$ . No entanto, ao utilizar esse método precisamos que  $\sum_{k=0}^{\infty} \gamma_k = \infty$  para termos garantia de convergência [7, Section 1.2].

Outra estratégia presente na literatura e que iremos utilizar é a *busca linear exata* que consiste em escolher, garantindo  $x_k + \gamma_k d_k \in C$ ,

$$\gamma_k = \arg \min_{\gamma > 0} \phi(\gamma), \quad (7)$$

em que  $\phi(\gamma) = f(x_k + \gamma d_k)$ . Esse valor de  $\gamma_k$  é o melhor possível, no sentido de que no ponto correspondente obtemos o menor valor de  $f$  entre todos os pontos de forma  $x_k + \hat{\gamma}_k d_k$ ,  $\hat{\gamma} \geq 0$ , e que pertencem a  $C$ . Em geral, resolver (7) pode ser difícil [34, Seção 3.1], mas há casos que isso pode ser feito de maneira eficiente.

Por exemplo, considere  $f(x) = \frac{1}{2}x^T Qx - q^T x$ , em que  $Q \in \mathbb{R}^{n \times n}$  é uma matriz simétrica definida positiva<sup>1</sup> e  $q \in \mathbb{R}^n$ . Encontrar o minimizador global de  $\phi(\gamma)$  equivale a encontrar  $\gamma_k$  tal que  $\phi'(\gamma) = \nabla f(x_k + \gamma d_k)^T d_k = 0$ . Usando o fato de que  $\nabla f(x) = Qx - q$  e que  $Q$  é uma matriz definida positiva, temos que  $\gamma_k$  que satisfaz (7) é dado por:

$$\gamma_k = \frac{-\nabla f(x_k)^T d_k}{d_k^T Q d_k}.$$

Ao aplicarmos os resultados desta seção para o caso em que o conjunto  $C$  é convexo, temos algumas especificações interessantes, as quais elucidamos na próxima seção.

### 2.3 OTIMIZAÇÃO SOBRE CONJUNTOS CONVEXOS

Os problemas de otimização que trataremos nessa tese, em sua maioria, terão o formato

$$\begin{aligned} \min_{x \in \mathbb{R}^m} \quad & f(x) \\ \text{s. a} \quad & x \in C \end{aligned} \quad (8)$$

em que  $C \subset \mathbb{R}^m$  é não-vazio, fechado e convexo, e  $f : \mathbb{R}^m \rightarrow \mathbb{R}$  é uma função continuamente diferenciável com gradiente  $L$ -Lipschitz, isto é,  $\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$ , para todo  $x, y \in \mathbb{R}^m$ , com  $L \geq 0$ .

Perceba que o problema (5) pode ser descrito por (8), em que  $C = \{x \in \mathbb{R}^m \mid h(x) = 0, g(x) \leq 0\}$ , desde que as funções  $h_i(x)$  sejam afins e as funções  $g_j(x)$  sejam convexas.

Ao minimizar uma função continuamente diferenciável sobre um conjunto convexo, podemos deduzir a seguinte condição necessária de otimalidade.

**Teorema 2.19** ([7, Proposition 2.1.2]). *Sejam  $C \subset \mathbb{R}^m$ , não-vazio, convexo e fechado,  $f : \Omega \rightarrow \mathbb{R}$  uma função continuamente diferenciável em um aberto  $\Omega$  contendo  $C$ . Se  $x^*$  é minimizador local de  $f$  em  $C$ , então:*

$$\nabla f(x^*)^T (x - x^*) \geq 0, \quad \forall x \in C. \quad (9)$$

<sup>1</sup> Uma matriz simétrica  $Q$  é definida positiva se  $x^T Qx > 0$ , para todos vetores não nulos  $x \in \mathbb{R}^n$  [7].

*Observação 2.20.* Se  $f$  é convexa, a condição acima assegura que  $x^*$  é um minimizador global de  $f$  em  $C$ .

Inspirado no Teorema 2.19, dizemos que  $x^*$  é um *ponto estacionário* para (8) se para  $x^* \in C$  se verifica (9). Perceba que esta definição de estacionariedade implica que a partir de  $x^* \in C$  não existe direção factível e de descida.

*Observação 2.21.* O Teorema 2.19 apresenta condições necessárias e suficientes para um problema de otimização convexa suave (ou seja, quando  $C$  é convexo e  $f$  é diferenciável e convexa). Além disso, as propriedades de projeção dadas pelo Teorema 2.9 nos permitem demonstrar que um ponto  $x^*$  é estacionário para (8) se, e somente se,  $x^* = P_C(x^* - \lambda \nabla f(x^*))$  para  $\lambda > 0$  [7, Section 2.3.1]. Este resultado será importante para métodos de gradiente projetado.

### 2.3.1 Métodos do tipo Frank-Wolfe

Nessa seção revisamos métodos do tipo Frank-Wolfe (FW) [15, 23, 35, 45] (conhecidos também como métodos de gradiente condicional), que fazem uso de um *oráculo linear* em sua iteração. Por oráculo linear, queremos dizer que minimizamos um funcional linear restrito a  $C$  para determinar o novo iterado. Esses métodos são de primeira ordem, ou seja, resolvem o problema (8) fazendo uso apenas da primeira derivada da função  $f$ .

O método de FW clássico está presente na literatura para solucionar o problema (8), no caso em que  $f$  é *convexa* e continuamente diferenciável e  $C$  é um conjunto não-vazio, convexo e *compacto*. Além disso, nessa seção veremos uma versão “acelerada” de FW, denominada Away Step Frank-Wolfe (ASFW). A partir daqui consideremos que  $f$  de (8) é uma função continuamente diferenciável com gradiente  $L$ -Lipschitz e convexa.

#### 2.3.1.1 Frank-Wolfe

O método de FW clássico, que está sintetizado no Algoritmo 1, tenta resolver o problema (8) com  $f$  convexa, através de uma sequência de subproblemas da forma

$$\begin{aligned} \min_{x \in \mathbb{R}^m} \quad & \nabla f(x_k)^T (x - x_k) \\ \text{s. a} \quad & x \in C, \end{aligned} \tag{10}$$

em que  $x_k$  denota o iterado corrente. O subproblema (10) é inspirado em uma aproximação linear da função objetivo em torno de  $x_k$ . Perceba que no caso de  $C$  ilimitado, o subproblema (10) pode não ter solução. Assim, assumiremos que  $C$  além de convexo, é também compacto<sup>2</sup>. Se  $\bar{x}_k$  denota uma solução de (10), a direção  $d_k := \bar{x}_k - x_k$  é uma direção factível e de descida para  $f$  a partir de  $x_k$ , e o método de FW prossegue com o novo iterado dado por  $x_{k+1} = x_k + \gamma_k d_k$  em que  $\gamma_k \in (0, 1]$ .

<sup>2</sup> No problema de inclusão que estamos interessados,  $\text{conv}(\mathcal{A})$  é sempre convexo e compacto, já que  $\mathcal{A}$  é formado por uma quantidade finita de pontos.

Uma das principais vantagens desse método é que o subproblema (10), em muitos casos, admite uma fórmula fechada, ou ainda tem um custo computacional menor do que resolver subproblemas quadráticos (como é o caso dos métodos do gradiente projetado [48] e do gradiente proximal [50]) [35]. Um exemplo disso é quando  $C = \Delta_m := \{x \in \mathbb{R}^m \mid e^T x = 1, x \geq 0\}$ , pois nesse caso, uma solução para (10) é dada por

$$\bar{x}_k = e_i, i \in \arg \min_{1 \leq i \leq m} (\nabla f(x_k))_i,$$

que pode ser calculado em  $\mathcal{O}(m)$  operações, enquanto a projeção sobre esse conjunto tem custo de  $\mathcal{O}(m \log m)$  [16].

Além disso, em casos particulares, por exemplo quando  $C = \text{conv}(\mathcal{A})$ , FW mantém as iterações como combinações convexas de poucos pontos de  $\mathcal{A}$ , o que é uma característica interessante em problemas que requerem soluções esparsas [14].

---

**Algoritmo 1: FRANK-WOLFE**


---

**Dados:**  $x_0 \in C, \epsilon > 0$   
**1 para**  $k = 0, 1, 2, \dots$  **faça**  
**2**     Encontre  $\bar{x}_k \in \arg \min \{\nabla f(x_k)^T (x - x_k) \mid x \in C\}$ .  
**3**      $d_k \leftarrow \bar{x}_k - x_k$ .  
**4**     **se**  $\nabla f(x_k)^T d_k \geq -\epsilon$  **então**  
**5**         | pare  
**6**     **fim**  
**7**     Determine o tamanho de passo  $\gamma_k \in (0, 1]$   
**8**      $x_{k+1} \leftarrow x_k + \gamma_k d_k$   
**9 fim**

---

Vejam agora alguns resultados sobre taxa de convergência (global) e complexidade de iteração de FW. Seja  $x^*$  uma solução de (8). Como  $f$  é convexa e diferenciável, temos que

$$h(x_k) := f(x_k) - f(x^*) \leq -\nabla f(x_k)^T (x^* - x_k) \leq -\min_{x \in C} \nabla f(x_k)^T (x - x_k) =: g(x_k),$$

ou seja,  $g(x_k)$  fornece um limite superior para o *gap primal*  $h(x_k)$ , a cada iteração do Algoritmo 1, mesmo não conhecendo o valor ótimo  $f(x^*)$  explicitamente. Isto justifica o critério de parada no passo 4.

Mais ainda, como  $f$  tem gradiente Lipschitz com constante  $L > 0$ , temos que

$$\begin{aligned} f(x_{k+1}) &= f(x_k + \gamma_k d_k) \leq f(x_k) + \gamma_k \nabla f(x_k)^T d_k + \frac{L}{2} \gamma_k^2 \|d_k\|^2 \\ &\leq f(x_k) - \gamma_k g(x_k) + \gamma_k^2 \frac{L}{2} D^2, \end{aligned}$$

em que  $D := \max\{\|x - y\| \mid x, y \in C\}$  é o diâmetro de  $C$ . Por essa inequação e a definição de  $h(x_k)$  obtemos

$$h(x_{k+1}) \leq (1 - \gamma_k)h(x_k) + \gamma_k^2 \frac{L}{2} D^2. \quad (11)$$

A partir da desigualdade (11), utilizando uma escolha de tamanho de passo  $\gamma_k$  específica, é possível mostrar que  $h(x_k) = \mathcal{O}(1/k)$ . Com efeito, considere  $\gamma_k = \frac{2}{k+2}$ . Iremos mostrar por indução que

$$h(x_k) \leq \frac{2LD^2}{k+2}. \quad (12)$$

Para  $k = 0$ ,  $\gamma_0 = \frac{2}{0+2} = 1$  e diretamente de (11) temos que  $h(x_1) \leq \frac{L}{2} D^2 \leq \frac{2LD^2}{3}$ . Utilizando (12) como hipótese indutiva, temos

$$\begin{aligned} h(x_{k+1}) &\leq (1 - \gamma_k)h(x_k) + \gamma_k^2 \frac{L}{2} D^2 \leq \left(1 - \frac{2}{k+2}\right) \frac{2LD^2}{k+2} + \left(\frac{2}{k+2}\right)^2 \frac{L}{2} D^2 \\ &= \left(\frac{k}{(k+2)^2} + \frac{1}{(k+2)^2}\right) 2LD^2 = \frac{k+1}{(k+2)^2} 2LD^2 \leq \frac{2LD^2}{k+1+2}, \end{aligned}$$

e assim a convergência *sublinear* de FW é estabelecida. Equivalentemente, são necessárias  $\mathcal{O}(1/\epsilon)$  iterações para que  $f(x_k) - f^* < \epsilon$ , em que  $f^*$  é o valor ótimo de (8). Note que se  $\bar{\gamma}_k$  é obtido por busca linear exata, a análise anterior ainda é válida porque  $f(x_k + \bar{\gamma}_k d_k) \leq f(x_k + \frac{2}{k+2} d_k)$ .

Infelizmente, este pior caso de complexidade de iteração  $\mathcal{O}(1/\epsilon)$  pode ser atingido quando  $x^*$  está na fronteira de  $C$ , como mostrado em [11].

No entanto, sob a hipótese adicional de  $f$  ser fortemente convexa, em certos casos é possível alcançar convergência linear. De fato, considerando esta hipótese e definindo  $e_k := x^* - x_k$ , temos que

$$f(x_k + \lambda e_k) - f(x_k) \geq \lambda \nabla f(x_k)^T e_k + \frac{\lambda^2}{2} \mu \|e_k\|^2 \geq -\frac{(\nabla f(x_k)^T \hat{e}_k)^2}{2\mu}, \quad \forall \lambda \in \mathbb{R},$$

em que  $\hat{e}_k := e_k / \|e_k\|$ . Em particular, para  $\lambda = 1$

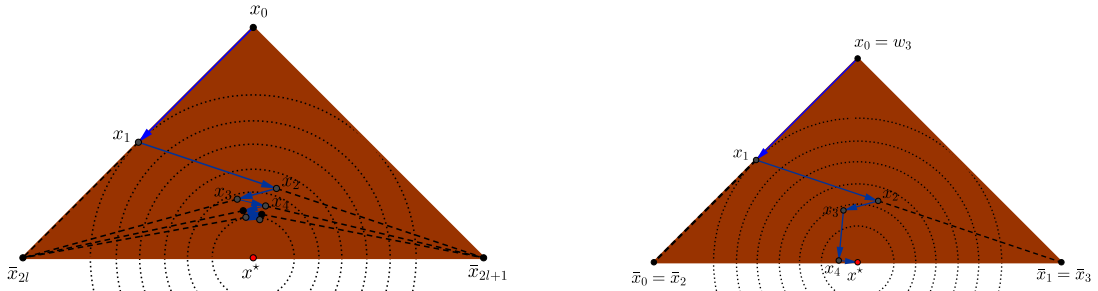
$$-h(x_k) \geq -\frac{(\nabla f(x_k)^T \hat{e}_k)^2}{2\mu}. \quad (13)$$

Por outro lado, usando novamente que  $f$  tem gradiente  $L$ -Lipschitz obtemos que

$$f(x_k + \gamma_k d_k) - f(x_k) \leq \gamma_k \nabla f(x_k)^T d_k + \frac{\gamma_k^2}{2} L \|d_k\|^2. \quad (14)$$

Seja  $\gamma_{\max} \in (0, 1]$  o maior tamanho de passo permitido. Se  $\gamma_k^* := -\frac{\nabla f(x_k)^T d_k}{L \|d_k\|^2} \leq \gamma_{\max}$ , tomando  $\gamma_k = \gamma_k^*$  temos que

$$\gamma_k \nabla f(x_k)^T d_k + \frac{\gamma_k^2}{2} L \|d_k\|^2 = -\frac{(\nabla f(x_k)^T \hat{d}_k)^2}{2L},$$



(a) Fenômeno de zigzag no Frank-Wolfe . (b) Iterações do Away-Step Frank-Wolfe.

Figura 2 – Exemplo de iterações FW e ASFW.

com  $\hat{d}_k := d_k / \|d_k\|$ . Note que, juntamente com (14), obtemos

$$h(x_k) - h(x_{k+1}) \geq \frac{(\nabla f(x_k)^T \hat{d}_k)^2}{2L}.$$

Este resultado, junto com (13), implica que

$$h(x_{k+1}) \leq \left[ 1 - \frac{\mu}{L} \left( \frac{\nabla f(x_k)^T \hat{d}_k}{\nabla f(x_k)^T \hat{e}_k} \right)^2 \right] h(x_k). \quad (15)$$

De (15), fica claro que a convergência linear dependerá de nossa capacidade de limitar a razão  $(\nabla f(x_k)^T \hat{d}_k) / (\nabla f(x_k)^T \hat{e}_k)$ . Isso pode ser feito em determinadas situações para o FW clássico ou variantes do FW usando diferentes direções de busca  $d_k$ .

Por exemplo, em [31, Theorem 2], Guélat e Marcotte mostram que se  $x^*$  está no interior relativo de  $C$ , com distância até a fronteira de  $C$  de pelo menos  $\rho > 0$  então, para  $k$  suficientemente grande,  $\nabla f(x_k)^T d_k \leq -(\rho/2) \|\nabla f(x_k)\|$  e  $\gamma_k^* \leq 1$ , o que implica que

$$h(x_{k+1}) \leq \left[ 1 - \frac{\mu\rho^2}{LD^2} \right] h(x_k). \quad (16)$$

Assim, temos convergência linear do método dependendo apenas das características da função  $f$  e da geometria do conjunto  $C$ .

No entanto, para o Frank-Wolfe clássico, se  $x^*$  está na fronteira de  $C$ , então  $\nabla f(x_k)^T \hat{d}_k$  pode ficar arbitrariamente próximo de zero e o método pode apresentar um comportamento de zig-zag (veja Figura 2a).

A seguir, discutiremos uma variante do método de Frank-Wolfe, chamada Away Step Frank-Wolfe (ASFW) que tem convergência *linear* independentemente da localização de  $x^* \in C$ .

### 2.3.1.2 Away-Step Frank-Wolfe

O algoritmo ASFW está resumido no Algoritmo 2, que foi baseado em [45, Alg. 1]. Fornecemos a seguir uma discussão sobre suas principais propriedades. Para tal algoritmo assumamos que  $C := \text{conv}(\mathcal{A})$ .

Seja  $V_C$  o conjunto dos pontos extremos de  $C$ . Em cada iteração  $k$  do Algoritmo 2, o iterado atual  $x_k$  é uma combinação convexa de elementos de  $\mathcal{A}$ :

$$x_k = \sum_{v \in \mathcal{A}} \alpha_v^k v. \quad (17)$$

Seja  $\alpha^k \in \Delta_n := \{x \in \mathbb{R}^n \mid e^T x = 1, x \geq 0\}$  um vetor contendo os coeficientes de tal combinação e defina  $U^k := \{v \in \mathcal{A} \mid \alpha_v^k > 0\}$ , isto é, o conjunto dos elementos de  $\mathcal{A}$  para os quais os coeficientes  $\alpha_v^k$  em (17) são estritamente positivos. Sem perda de generalidade, assumamos que os elementos de  $\mathcal{A}$  sejam ordenados de tal forma que  $x_0$  seja o primeiro: então  $\alpha^0 = e_1$ , é o primeiro vetor canônico de  $\mathbb{R}^n$ .

---

**Algoritmo 2:** Away-Step Frank-Wolfe (ASFW)

---

**Dados:**  $x_0 \in \mathcal{A}, \epsilon > 0, U^0 = \{x_0\}, \alpha^0 = e_1 \in \Delta_n$

1 **para**  $k = 0, 1, 2, \dots$  **faça**

2     Seja  $\bar{x}_k \in \arg \min_{x \in \mathcal{A}} \nabla f(x_k)^T (x - x_k)$

3      $d_k^{FW} \leftarrow \bar{x}_k - x_k$

4     Seja  $w_k \in \arg \max_{x \in U^k} \nabla f(x_k)^T (x - x_k)$

5      $d_k^A \leftarrow x_k - w_k$

6     **se**  $\nabla f(x_k)^T d_k^{FW} \geq -\epsilon$  **então**

7         | pare e retorne  $x_k$

8     **fim**

9     **se**  $\nabla f(x_k)^T d_k^{FW} \leq \nabla f(x_k)^T d_k^A$  **então**

10         |  $d_k \leftarrow d_k^{FW}, \gamma_{\max} \leftarrow 1$

11     **senão**

12         |  $d_k \leftarrow d_k^A, \gamma_{\max} \leftarrow \alpha_{w_k} / (1 - \alpha_{w_k})$

13     **fim**

14     Determine o tamanho de passo  $\gamma_k \in (0, \gamma_{\max}]$

15      $x_{k+1} \leftarrow x_k + \gamma_k d_k$ .

16      $U^{k+1} \leftarrow \{v \in \mathcal{A} \mid \alpha_v^{k+1} > 0\}$ .

17 **fim**

---

Em cada iteração do ASFW, duas direções de busca são computadas: a direção FW  $d_k^{FW} = \bar{x}_k - x_k$  e uma *direção away*  $d_k^A = x_k - w_k$ , em que  $\bar{x}_k$  é uma solução de (10) e  $w_k$  é tal que  $\nabla f(x_k)^T (w_k - x_k) \geq \nabla f(x_k)^T (u - x_k), \forall u \in U^k$ . Se  $\nabla f(x_k)^T d_k^{FW} \leq \nabla f(x_k)^T d_k^A$ , é escolhida a direção FW clássica e o tamanho de passo máximo é  $\gamma_{\max} = 1$ . Caso contrário,  $\gamma_{\max} = \alpha_{w_k} / (1 - \alpha_{w_k})$  fornece um limitante superior *conservador* para o tamanho do passo na direção  $d_k^A$  garantir a factibilidade do novo iterado  $x_{k+1}$ . De fato, note que  $x_k$  é uma combinação convexa dos pontos de  $U^k$ , ou seja,  $x_k = \sum_{u \in U^k} \alpha_u u$  com  $\sum_{u \in U^k} \alpha_u = 1$ , e



que  $w_k \in U^k$ . Sendo assim, para  $\gamma_{\max} := \alpha_{w_k}/(1 - \alpha_{w_k})$ , temos que

$$\begin{aligned} x_{k+1} &:= x_k + \frac{\alpha_{w_k}}{1 - \alpha_{w_k}}(x_k - w_k) \\ &= \frac{x_k - \alpha_{w_k}w_k}{1 - \alpha_{w_k}} \\ &= \frac{1}{1 - \alpha_{w_k}} \sum_{u \in U^k \setminus \{w_k\}} \alpha_u u \in C = \text{conv}(\mathcal{A}), \end{aligned}$$

pois  $U_k \subset C$  e  $\sum_{u \in U^k \setminus \{w_k\}} \alpha_u = 1 - \alpha_{w_k}$ .

*Observação 2.22.* Quando  $d_k = d_k^A$  e  $\gamma_k = \gamma_{\max}$ , então  $U^{k+1} = U^k \setminus \{w_k\}$ . Se o tamanho do passo for menor que  $\gamma_{\max}$ ,  $U^k$  permanece inalterado. Quando  $d_k = d_k^{FW}$ , se  $\gamma_k = 1$ , então  $U^{k+1} = \{\bar{x}_k\}$ , caso contrário  $U^{k+1} = U^k \cup \{\bar{x}_k\}$ , em que  $\bar{x}_k$  é dado no Passo 2 do Algoritmo 2. Isso significa que a cardinalidade de  $U^k$  diminui ou aumenta em um a cada iteração. Portanto, se  $s^k$  é o número de iterações, até a iteração  $k$ , em que  $d_k = d_k^A$  e  $\gamma_k = \gamma_{\max}$ , e  $\ell^k$  é o número de iterações em que um elemento foi adicionado a  $U^k$ , temos  $s^k + \ell^k \leq k - 1$ ,  $s^k \leq \ell^k$  e, portanto,  $s^k \leq (k - 1)/2$ .

Vamos definir  $d_k^P := \bar{x}_k - w_k$ , que é conhecido na literatura como direção *pairwise*; veja [45]. Da definição de  $d_k^P$  e do Passo 9 do Algoritmo 2 temos que

$$-\nabla f(x_k)^T d_k^P \leq -2\nabla f(x_k)^T d_k.$$

Além disso, utilizando  $d_k^P$  podemos estabelecer um limite para a razão

$$\frac{\nabla f(x_k)^T \hat{d}_k^P}{\nabla f(x_k)^T \hat{e}_k}$$

em (15). Para isso, consideramos o próximo lema, que é válido se o conjunto convexo  $C$  for um poliedro compacto, ou seja,  $C := \{x \in \mathbb{R}^m \mid a_i^T x \leq b_i, i = 1, \dots, \ell\}$ . A seguir, para  $U \subset \mathbb{R}^m$ , defina  $\mathcal{I}_C(U)$  o conjunto de restrições do poliedro  $C$  que são ativas em todos os pontos de  $U$ , isto é,  $\mathcal{I}_C(U) := \{i \in \{1, \dots, \ell\} \mid a_i^T u = b_i, \forall u \in U\}$ .

**Lema 2.23** ([6, Lemma 3.1]). *Seja  $C := \{x \in \mathbb{R}^m \mid a_i^T x \leq b_i, i = 1, \dots, \ell\}$  um poliedro compacto com  $V_C$  como o conjunto de seus pontos extremos,  $c \in \mathbb{R}^m$ , e considere  $U \subset V_C$ . Se existe  $z \in \mathbb{R}^m$  tal que  $a_i^T z \leq 0$ , para todo  $i \in \mathcal{I}_C(U)$ , e  $c^T z > 0$ , então*

$$\max_{p \in V_C, u \in U} c^T(p - u) \geq \frac{\Omega_C}{m + 1} \frac{c^T z}{\|z\|},$$

em que

$$\Omega_C := \min_{v \in V_C, i \in \{1, \dots, \ell\} \mid b_i > a_i^T v} \frac{b_i - a_i^T v}{\|a_i\|}.$$

O Lema 2.23 pode ser aplicado para  $C = \text{conv}(\mathcal{A})$ , pois  $\text{conv}(\mathcal{A})$  é um poliedro compacto [7], embora normalmente não conheçamos explicitamente as desigualdades lineares que o definem. Assim, se assumirmos que  $\bar{x}_k \in V_C \subset \mathcal{A}$  no Algoritmo 2, podemos

escrever

$$-\nabla f(x_k)^T d_k^P = -\nabla f(x_k)^T (\bar{x}_k - w_k) = \max_{p \in V_C, u \in U^k} -\nabla f(x_k)^T (p - u),$$

e para  $z := e_k = x^* - x_k$  e  $c := -\nabla f(x_k)$ , podemos aplicar o Lema 2.23 para obter

$$-2\nabla f(x_k)^T d_k \geq -\nabla f(x_k)^T d_k^P \geq \frac{\Omega_C}{m+1} (-\nabla f(x_k))^T \hat{e}_k > 0.$$

Esta última desigualdade mostra que, em cada iteração do ASFW para a qual  $\gamma_k^* \leq \gamma_{\max}$ , temos

$$h(x_{k+1}) \leq \left[ 1 - \frac{\mu}{4L} \left( \frac{\Omega_C}{D(m+1)} \right)^2 \right] h(x_k).$$

Quando  $d_k = d_k^A$  e  $\gamma_k^* > \gamma_{\max}$ , só podemos garantir que  $h(x_{k+1}) \leq h(x_k)$ . No entanto, como discutido anteriormente (veja Observação 2.22), esse tipo de iteração não pode ocorrer com muita frequência. Portanto, para o ASFW temos

$$h(x_k) \leq h(x_0) \left[ 1 - \frac{\mu}{4L} \left( \frac{\Omega_C}{D(m+1)} \right)^2 \right]^{(k-1)/2}, \quad \forall k \in \mathbb{N}, \quad (18)$$

que implica convergência global linear (já que  $\Omega_C \leq D$  e  $\mu \leq L$ ). Resumimos a discussão acima, que combina a análise de [45] com o Lema 2.23, no seguinte resultado.

**Teorema 2.24.** *Seja  $f$   $\mu$ -fortemente convexa,  $\nabla f$   $L$ -Lipschitz,  $C$  um poliedro compacto (com diâmetro  $D$ ) e seja  $f(x^*)$  o valor ótimo para (8). Seja  $(x_k)_{k=0}^\infty$  a sequência gerada pelo ASFW para o problema (8) e suponha que  $\bar{x}_k \subset V_C$ . Então, para todo  $k \geq 1$ ,*

$$f(x_k) - f(x^*) \leq (1 - \eta)^{(k-1)/2} (f(x_0) - f(x^*)),$$

em que  $\eta := \frac{\mu}{4L} \left( \frac{\Omega_C}{D(m+1)} \right)^2$ , com  $\Omega_C$  como no Lema 2.23.

### 2.3.2 Métodos do tipo Gradiente Projetado

Nessa seção vamos rever brevemente outro método de primeira ordem bem conhecido na literatura: o método do Gradiente Projetado. Iremos focar especificamente no uso deste método para resolver o problema (8) [48], bem como exploraremos uma versão mais recente [8, 10] do mesmo.

Dado um ponto factível  $x_k \in C$ , o método consiste em realizar uma busca linear ao longo da direção  $d_k = \bar{x}_k - x_k$ , em que  $\bar{x}_k = P_C(x_k - \nabla f(x_k))$ . Não é difícil mostrar que, se  $x_k$  não é estacionário, então  $d_k$  é uma direção factível e de descida, e que  $x_k + \gamma d_k \in C$ , para todo  $\gamma \in [0, 1]$ . Note que se  $x_k - \nabla f(x_k) \in C$ , a direção de busca é a mesma do tradicional método do gradiente para otimização irrestrita [48].

O Algoritmo 3 sintetiza o método de Gradiente Projetado com busca linear.

**Algoritmo 3: GRADIENTE PROJETADO**


---

**Dados:**  $x_0 \in C, \varepsilon > 0$ .

- 1 **para**  $k = 0, 1, 2, \dots$  **faça**
- 2     Seja  $\bar{x}_k = P_C(x_k - \nabla f(x_k))$
- 3      $d_k \leftarrow \bar{x}_k - x_k$
- 4     **se**  $\|d_k\| < \varepsilon$  **então**
- 5         | pare
- 6     **fim**
- 7     Determine o tamanho de passo  $\gamma_k \in (0, 1]$  ;
- 8      $x_{k+1} \leftarrow x_k + \gamma_k d_k$
- 9 **fim**

---

Para  $x_k \in C$  dado, se  $d_k = 0$ , temos então, pela Observação 2.21 que  $x_k$  é estacionário. Isso justifica o critério de parada no Passo 4 do Algoritmo 3.

O principal custo por iteração dos métodos de gradiente projetado é o custo de calcular a projeção  $P_C(\cdot)$ . O cálculo da projeção pode ser simples para alguns conjuntos como caixas, bolas, hiperplanos, semi-espacos, e arbitrariamente complicada em outros casos. Uma alternativa para quando a projeção em  $C$  é muito cara é o algoritmo de Frank-Wolfe apresentado nas subseções 2.3.1.1 e 2.3.1.2.

Nas últimas décadas, estratégias não-monótonas para otimização não-linear começaram a se tornar populares [29, 30, 56] e, juntamente com ideias de *scaling*, estão presentes em versões aprimoradas do método do gradiente projetado.

Um exemplo é o método Gradiente Projetado Espectral (ou simplesmente SPG, do inglês *Spectral Projected Gradient*), proposto por Birgin, Martínez e Raydan [8] e Birgin, Martínez e Raydan [9], que usa o *scaling* espectral de Barzilai e Borwein [4] da direção do menos gradiente e busca linear não monótona [17]. O Algoritmo 4 resume o método do Gradiente Projetado Espectral como em [10].

A iteração do método SPG tem a forma  $x_{k+1} = x_k + \gamma_k d_k$ , em que a direção  $d_k$  é definida como  $d_k := P_C(x_k - \lambda_k \nabla f(x_k)) - x_k$ . Aqui,  $\lambda_k$  é o parâmetro espectral

$$\lambda_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T u_{k-1}}, \quad (19)$$

com  $s_{k-1} := x_k - x_{k-1}$  e  $u_{k-1} := \nabla f(x_k) - \nabla f(x_{k-1})$ . Em verdade, o Algoritmo 4 usa uma versão de (19) com salva-guardas para evitar que tal parâmetro fique muito grande ou muito próximo de zero.

Em vez de impor uma diminuição suficiente da função objetivo a cada iteração, uma característica do SPG é empregar uma busca linear não monótona para favorecer a aceitação do passo completo  $\gamma_k = 1$ , enquanto ainda garante a convergência global [9]. Por exemplo, a busca linear não monótona proposta por Grippo, Lampariello e Lucidi [30] requer apenas uma diminuição em relação ao maior valor funcional das últimas  $M$

iterações. Mais especificamente, o tamanho de passo  $\gamma$  deve satisfazer

$$f(x_k + \gamma d_k) \leq f_{\max} + \eta \gamma \nabla f(x_k)^T d_k,$$

em que  $f_{\max} := \max\{f(x_{k-j}) \mid 0 \leq j \leq \min\{k, M-1\}\}$  e  $\eta \in (0, 1)$ .

Em [29] é demonstrado que para  $f$  convexa, em  $\mathcal{O}(1/\varepsilon)$  iterações do Gradiente Projetado Espectral com busca não-monótona obtemos,  $f(x_k) - f(x^*) \leq \varepsilon$ , em que  $x^*$  é solução de (8).

---

**Algoritmo 4:** Gradiente Projetado Espectral

---

**Dados:**  $x_0 \in C, \epsilon > 0, M \geq 1, \eta \in (0, 1), 0 < \lambda_{\min} \leq \lambda_{\max} < \infty$  e  $\lambda_0 \in [\lambda_{\min}, \lambda_{\max}]$ .

- 1 **para**  $k = 0, 1, 2, \dots$  **faça**
- 2      $\bar{x}_k = P_C(x_k - \lambda_k \nabla f(x_k));$
- 3      $d_k \leftarrow \bar{x}_k - x_k.$
- 4     **se**  $\|d_k\| < \epsilon$  **então**
- 5         Pare: retorne  $x_k$ .
- 6     **fim**
- 7     Compute  $f_{\max} = \max\{f(x_{k-j}) \mid 0 \leq j \leq \min\{k, M-1\}\}$  e seja  $\gamma \leftarrow 1$ .
- 8     **se**  $f(x_k + \gamma d_k) \leq f_{\max} + \eta \gamma \nabla f(x_k)^T d_k$  **então**
- 9          $\gamma_k \leftarrow \gamma.$
- 10    **senão**
- 11          $\gamma \leftarrow \gamma/2$  e vá para o passo 8.
- 12    **fim**
- 13     $x_{k+1} \leftarrow x_k + \gamma_k d_k.$
- 14     $s_k \leftarrow x_{k+1} - x_k$
- 15     $u_k \leftarrow \nabla f(x_{k+1}) - \nabla f(x_k).$
- 16    **se**  $s_k^T u_k \leq 0$  **então**
- 17          $\lambda_{k+1} \leftarrow \lambda_{\max}.$
- 18    **senão**
- 19          $\lambda_{k+1} \leftarrow \max\{\lambda_{\min}, \min\{s_k^T s_k / s_k^T u_k, \lambda_{\max}\}\}.$
- 20    **fim**
- 21 **fim**

---

No próximo capítulo, investigaremos como os métodos do tipo Frank-Wolfe e Gradiente Projetado podem ser adaptados para resolver de forma eficiente o problema de decisão CHMP.

### 3 MÉTODOS DE PRIMEIRA ORDEM PARA O PROBLEMA DE INCLUSÃO NO ENVOLTÓRIO CONVEXO

Neste capítulo discutiremos como métodos de primeira ordem podem ser eficientes na resolução do problema de inclusão no envoltório convexo (CHMP). Mais especificamente, nas Seções 3.1 e 3.2 apresentamos o problema de inclusão no envoltório convexo e fornecemos uma descrição formal do Algoritmo do Triângulo, juntamente com seus principais resultados de convergência e complexidade de iteração. Na Seção 3.3, discutimos as semelhanças entre os algoritmos do Triângulo e Frank-Wolfe aplicados ao CHMP. Mostramos que FW com um critério de parada específico corresponde a uma variante gananciosa do Algoritmo do Triângulo. Por outro lado, veremos também que o Algoritmo do Triângulo pode ser visto como uma versão inexata de FW aplicada ao CHMP. A Seção 3.4 discute a aplicação de métodos de gradiente projetado ao problema de inclusão e na Seção 3.5 propomos novos critérios de parada para Frank-Wolfe e métodos de gradiente projetado, específicos para o CHMP.

Os experimentos numéricos relatados na Seção 3.6 mostram que os métodos de primeira ordem equipados com critérios de parada adequados podem ser competitivos com o Algoritmo do Triângulo e variantes. Relatamos isso em problemas gerados artificialmente cobrindo diferentes casos (conforme descrito na Seção 3.6.1) com relação à geometria do envoltório convexo e à posição do ponto a ser consultado. Duas potenciais aplicações envolvendo o CHMP, uma no problema de factibilidade em PL e outra no problema de classificação de imagens, são detalhadas na Seção 3.6.2 e na Seção 3.6.3, respectivamente.

Como mencionado na introdução, os resultados deste capítulo foram recentemente publicados em artigo do periódico EJOR [22].

#### 3.1 O PROBLEMA DE INCLUSÃO NO ENVOLTÓRIO CONVEXO

Vamos iniciar essa seção recordando a definição do problema de inclusão no envoltório convexo que já foi vista na introdução.

**Definição 3.1.** Sejam  $\mathcal{A} := \{v_1, v_2, \dots, v_n\} \subset \mathbb{R}^m$  e  $p \in \mathbb{R}^m$  um ponto dado. O *problema de inclusão no envoltório convexo* (CHMP) consiste em determinar se

$$p \in \text{conv}(\mathcal{A}),$$

em que  $\text{conv}(\mathcal{A})$  denota o envoltório convexo de  $\mathcal{A}$ .

Seja  $A := [v_1 \ v_2 \ \dots \ v_n] \in \mathbb{R}^{m \times n}$  a matriz em que cada coluna é um dos  $n$  pontos de  $\mathcal{A}$ . Podemos ver que  $p \in \text{conv}(\mathcal{A})$  se e somente se  $p$  é uma combinação convexa das colunas de  $A$ . Relembre que o CHMP pode ser interpretado como o seguinte problema de factibilidade de programação linear:

$$\text{Existe algum } x \in \mathbb{R}^n \text{ tal que } Ax = p, e^T x = 1, x \geq 0? \quad (20)$$

A resposta afirmativa para (20) garante  $p \in \text{conv}(\mathcal{A})$ . Equivalentemente tal problema de factibilidade também pode ser resolvido pelo seguinte problema de programação quadrática

$$\min_{x \in \Delta_n} \frac{1}{2} \|Ax - p\|^2 =: \Phi(x), \quad (21)$$

em que  $\Delta_n := \{x \in \mathbb{R}^n \mid e^T x = 1, x \geq 0\}$  é o simplex unitário em  $\mathbb{R}^n$ . Claramente,  $p \in \text{conv}(\mathcal{A})$  quando o valor ótimo de (21) é zero. Além disso, se  $x$  é um ponto viável de (21), então  $y := Ax \in \text{conv}(\mathcal{A})$ . Portanto, outra possível formulação é dada por

$$\begin{aligned} \min_{y \in \mathbb{R}^m} \quad & \frac{1}{2} \|y - p\|^2 =: \Psi(y) \\ \text{s. a} \quad & y \in \text{conv}(\mathcal{A}), \end{aligned} \quad (22)$$

que atinge um valor ótimo zero se e somente se  $p \in \text{conv}(\mathcal{A})$ . Vale ressaltar que a solução de (22) é nada mais que a projeção de  $p$  sobre  $\text{conv}(\mathcal{A})$ . No entanto, não sabemos como obter tal projeção de forma direta/analítica pois, apesar de  $\text{conv}(\mathcal{A})$  ser um poliedro compacto, em geral, se quer conhecemos as desigualdades lineares que o definem (temos acesso apenas aos elementos de  $\mathcal{A}$ ).

Tendo as formulações (20), (21) e (22) pode-se simplesmente aplicar métodos clássicos de programação linear ou quadrática para resolver o CHMP. Mesmo assim, em [37], inspirado por ideias geométricas, Kalantari propôs um algoritmo específico para o CHMP, chamado Algoritmo do Triângulo (TA), que mostra bons resultados numéricos [47] em comparação ao método Simplex aplicado a (20) ou a métodos do tipo Frank-Wolfe e Gradiente Projetado aplicados ao problema (21).

### 3.2 ALGORITMO DO TRIÂNGULO

O Algoritmo do Triângulo é baseado em um teorema de alternativas conhecido por *Teorema de dualidade de distâncias*. Antes de enunciarmos tal teorema, apresentaremos um resultado auxiliar.

**Lema 3.2.** *Sejam  $\mathcal{A} = \{v_1, v_2, \dots, v_n\} \subset \mathbb{R}^m$  e  $p \in \mathbb{R}^m$ ,  $p \notin \mathcal{A}$ . Temos que  $p \notin \text{conv}(\mathcal{A})$  se, e somente se, existe  $p' \in \text{conv}(\mathcal{A})$  tal que  $d(v_j, p) > d(v_j, p')$ ,  $\forall v_j \in \mathcal{A}$ .*

Em [37] tal resultado é demonstrado usando o conceito de célula de Voronoi. No Apêndice A apresentamos uma demonstração alternativa baseada no Teorema da Projeção (Teorema 2.9), apresentada pela primeira vez em [21].

Uma consequência do Lema 3.2 é que se  $p \notin \text{conv}(\mathcal{A})$ , o hiperplano bissetor ortogonal ao segmento  $[p', p]$ , dado por

$$H = \left\{ x \in \mathbb{R}^m \mid (p - p')^T x = \frac{\|p\|^2 - \|p'\|^2}{2} \right\}, \quad (23)$$

separa estritamente  $p$  de  $\text{conv}(\mathcal{A})$ , como ilustrado na Figura 3. Por esse motivo chamaremos de *testemunha* um  $p' \in \text{conv}(\mathcal{A})$  tal que  $d(v_j, p) > d(v_j, p')$ ,  $\forall v_j \in \mathcal{A}$ . É oportuno pontuar

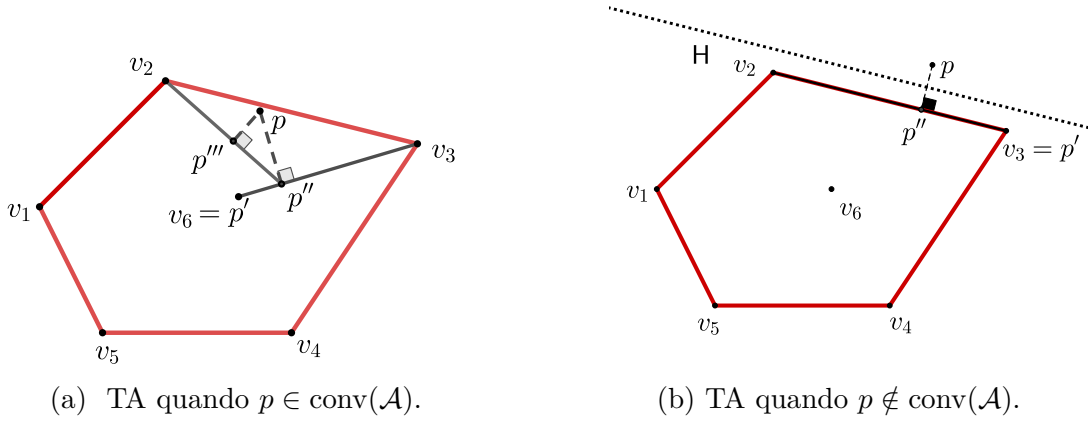


Figura 3 – Iterações do Algoritmo do Triângulo.

que quando  $p \notin \text{conv}(\mathcal{A})$ , a projeção de  $p$  sobre  $\text{conv}(\mathcal{A})$  é uma testemunha, mas não é a única, isto é, podem existir outros  $p' \in \text{conv}(\mathcal{A})$  com a propriedade que acabamos de mencionar. Além disso, podemos mostrar que quando  $p \notin \text{conv}(\mathcal{A})$ , para uma testemunha  $p'$ , temos

$$\frac{1}{2}\|p' - p\| \leq \Delta \leq \|p' - p\|, \quad (24)$$

em que  $\Delta := \min\{d(v, p) \mid v \in \text{conv}(\mathcal{A})\}$ . Assim, a distância de  $p$  a  $\text{conv}(\mathcal{A})$  é aproximada por um fator de no máximo 2. Caso contrário, temos a garantia de que para cada  $p' \in \text{conv}(\mathcal{A})$ , existe  $v \in \mathcal{A}$  tal que  $v$  está mais próximo de  $p$  do que de  $p'$ , o que dá origem ao seguinte teorema.

**Teorema 3.3** (Dualidade de Distâncias). *Sejam  $\mathcal{A} = \{v_1, v_2, \dots, v_n\} \subset \mathbb{R}^m$  e  $p \in \mathbb{R}^m$ . Exatamente uma das seguintes condições é satisfeita*

- (a)  $p \notin \text{conv}(\mathcal{A})$ ;
- (b) Para todo  $p' \in \text{conv}(\mathcal{A})$ , existe  $v_j \in \mathcal{A}$  tal que  $d(v_j, p) \leq d(v_j, p')$ .

Com base neste teorema descrevemos o Algoritmo do Triângulo, o qual a cada iteração seleciona  $v_i \in \mathcal{A}$ , tal que  $d(p', v_i) \geq d(p, v_i)$ . Tal  $v_i \in \mathcal{A}$  é chamado de *pivô*. Se um pivô não existir, podemos parar e declarar  $p \notin \text{conv}(\mathcal{A})$ , segundo o Teorema 3.3. Caso contrário, o próximo iterado é o ponto situado no segmento  $[p', v]$  que está mais próximo de  $p$ . Seja  $R := \max\{d(v_i, p) \mid v_i \in \mathcal{A}\}$ . As iterações continuam até  $\|p' - p\| \leq \varepsilon R$ , quando dizemos que  $p'$  é uma  $\varepsilon$ -solução, e o algoritmo para, informando  $p \in \text{conv}(\mathcal{A})$ . A Figura 3 ilustra esse procedimento (em que a sequência de iterados é denotada por  $p', p'', \dots$ ).

Este esquema iterativo é resumido no Algoritmo 5 e sua boa definição e correteude seguem do Teorema 3.3 juntamente com o Lema 3.2.

*Observação 3.4.* Nos casos em que  $p \in \text{conv}(\mathcal{A})$  um critério de parada natural seria  $d(p_k, p) \leq \varepsilon$ . No entanto, este critério de parada pode ser problemático quando o diâmetro de  $\text{conv}(\mathcal{A})$  é muito pequeno. Por exemplo, considere que  $\mathcal{A} \subset B(0, \varepsilon/2)$ . Suponha  $p \notin$

**Algoritmo 5:** ALGORITMO DO TRIÂNGULO (TA)

---

**Dados:**  $\mathcal{A} := \{v_1, \dots, v_n\}, p, R, \varepsilon \in (0, 1)$

- 1 Escolha  $p_0 \in \arg \min\{d(v_j, p) \mid v_j \in \mathcal{A}\}$
- 2 **para**  $k = 0, 1, 2, \dots$  **faça**
- 3     **se**  $d(p_k, p) < \varepsilon R$  **então**
- 4         | pare
- 5     **fim**
- 6     **se**  $d(v_j, p) > d(v_j, p_k), \forall j = 1, \dots, n$  **então**
- 7         | pare:  $p_k$  é uma  $p$ -testemunha,  $p \notin \text{conv}(\mathcal{A})$ .
- 8     **senão**
- 9         | Escolha um pivô:  $v_j \neq p_k$ , tal que  $d(v_j, p) \leq d(v_j, p_k)$ .
- 10    **fim**
- 11    Seja  $\bar{\gamma}_k \in \arg \min\{d(p, (1 - \gamma)p_k + \gamma v_j) \mid 0 \leq \gamma \leq 1\}$ .
- 12     $p_{k+1} \leftarrow (1 - \bar{\gamma}_k)p_k + \bar{\gamma}_k v_j$ .
- 13 **fim**

---

$\text{conv}(\mathcal{A})$ , mas  $p \in B(0, \varepsilon/2)$ . Assim,  $d(p, \text{conv}(\mathcal{A})) < \varepsilon$  e para todo  $p' \in \text{conv}(\mathcal{A})$  temos que  $d(p', p) < \varepsilon$ , isto é, o algoritmo pararia declarando  $p \in \text{conv}(\mathcal{A})$ , mas na verdade  $p \notin \text{conv}(\mathcal{A})$ . Assim, ao multiplicar  $\varepsilon$  por  $R = \max\{d(v_j, p) \mid v_j \in \mathcal{A}\}$  evitamos este equívoco.

Com o objetivo de facilitar algumas demonstrações, condensamos no lema a seguir algumas caracterizações equivalentes para um pivô que seguem de simples manipulações algébricas.

**Lema 3.5** ([21, Lema 2.4]). *Seja  $p_k \in \text{conv}(\mathcal{A})$ ,  $v_j \in \mathcal{A}$  e  $p \in \mathbb{R}^m$  o ponto a ser consultado. As seguintes afirmações são equivalentes:*

- (a)  $\|v_j - p\| \leq \|v_j - p_k\|$ ;
- (b)  $2v_j^T(p_k - p) \leq \|p_k\|^2 - \|p\|^2$ ;
- (c)  $(p_k - p)^T(v_j - p_k) \leq -\frac{1}{2}\|p_k - p\|^2$ ;
- (d)  $(p_k - p)^T(v_j - p) \leq \frac{1}{2}\|p_k - p\|^2$ .

Com o auxílio desse lema podemos perceber que o Algoritmo do Triângulo é um algoritmo de descida, que reduz a distância em relação a  $p$  a cada iteração, como mostra o seguinte resultado.

**Lema 3.6.** *Na  $k$ -ésima iteração do Algoritmo 5, se  $d(p_k, p) > \varepsilon R$  e se existe um pivô  $v_j$ , então  $\bar{\gamma}_k$  do passo 11 tem uma fórmula fechada dada por*

$$\bar{\gamma}_k = -\frac{(p_k - p)^T(v_j - p_k)}{\|v_j - p_k\|^2} \in (0, 1]. \quad (25)$$

Além disso,  $d(p_{k+1}, p) < d(p_k, p)$ .



*Demonstração.* Note que para calcular o minimizador irrestrito da função quadrática convexa  $\phi(\gamma) := d(p, (1 - \gamma)p_k + \gamma v_j)^2$  requerido no passo 11 basta derivar e igualar a 0. Fazendo isso temos (25). Vamos mostrar que  $\bar{\gamma}_k \in (0, 1]$ . Do Lema 3.5(c), temos que

$$\bar{\gamma}_k \geq \frac{1}{2} \frac{\|p_k - p\|^2}{\|v_j - p_k\|^2} > 0,$$

em que a última inequação segue da hipótese de que  $d(p_k, p) > \varepsilon R > 0$ . Por outro lado, da desigualdade de Cauchy-Schwarz, obtemos

$$\bar{\gamma}_k \leq \frac{\|p_k - p\|}{\|v_j - p_k\|} \leq \frac{\|p_k - p\|}{\|v_j - p\|} \leq 1,$$

em que a segunda inequação segue do passo 9 junto com o Lema 3.5(a). A terceira desigualdade é assegurada na primeira iteração pelo passo 1. Sua validade nas próximas iterações segue do fato de que  $d(p_{k+1}, p) < d(p_k, p)$ . Com efeito

$$\begin{aligned} d(p_{k+1}, p)^2 &= \|p_{k+1} - p\|^2 = \|(1 - \bar{\gamma}_k)p_k + \bar{\gamma}_k v_j - p\|^2 = \|\bar{\gamma}_k(v_j - p_k) + (p_k - p)\|^2 \\ &= \bar{\gamma}_k^2 \|v_j - p_k\|^2 + \bar{\gamma}_k 2(p_k - p)^T (v_j - p_k) + \|p_k - p\|^2 \\ &< \|p_k - p\|^2 = d(p_k, p)^2, \end{aligned} \quad (26)$$

em que a última desigualdade decorre do fato de que  $\bar{\gamma}_k > 0$  em (25) é o minimizador restrito da quadrática em (26). Sendo assim,  $d(p_{k+1}, p) < d(p_k, p)$  e  $0 < \bar{\gamma}_k \leq 1$  em cada iteração na qual  $d(p_k, p) > \varepsilon R$  e existe um pivô  $v_j$ .  $\square$

Como consequência do Lema 3.6, temos que a sequência de distâncias dada por  $\delta_k := d(p_k, p)$  é decrescente. Podemos de fato quantificar tal redução observando que as desigualdades  $d(p_k, p) \leq d(v, p) \leq d(v, p_k)$  implicam que os pontos  $p_k, p$  e  $v$  são não-colineares e, assim, formam um triângulo não degenerado, como representado na Figura 4a, em que o ângulo  $\theta_k := \angle pp_k v \in [0, \pi/2)$ . Assim, temos que

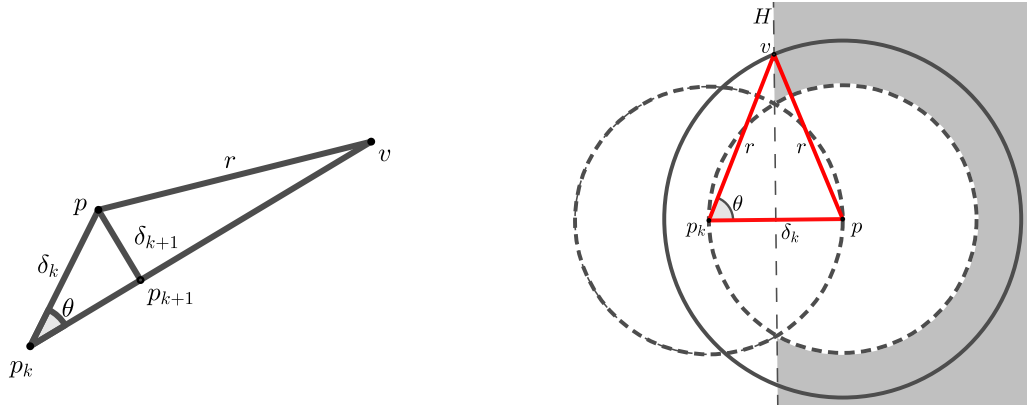
$$\|p_{k+1} - p\|^2 = (1 - \cos^2 \theta_k) \|p_k - p\|^2. \quad (27)$$

Tendo em vista a caracterização de pivô (Lema 3.5), podemos ver que se  $\|v - p\| = r$ ,  $\cos \theta_k$  é mínimo quando  $v$  está sobre o hiperplano  $H$  definido em (23). Neste caso,  $\cos \theta_k = \delta_k / 2r$ , conforme ilustração na Figura 4b.

Destas observações e de (27) temos o próximo teorema.

**Teorema 3.7** ([37, Theorem 8]). *Sejam  $p, p', v$  pontos distintos em  $\mathbb{R}^m$  e suponha que  $d(v, p) \leq d(v, p')$ . Seja  $p''$  o ponto no segmento  $[p', v]$  que está mais próximo de  $p$ . Defina  $\delta := d(p', p)$ ,  $\delta' := d(p'', p)$ ,  $r := d(v, p)$  e assumo  $\delta \leq r$ . Então,*

$$\delta' \leq \delta \sqrt{1 - \frac{\delta^2}{4r^2}}.$$



(a) Iteração do Algoritmo do Triângulo. (b) Pior caso para  $\cos \theta$  quando  $v$  é um pivô. A área sombreada representa a região onde um pivô pode ser encontrado.

Figura 4 – Motivação para a prova do Teorema 3.7.

Considerando o Algoritmo 5 e utilizando o teorema acima nós temos que

$$\delta_{k+1} \leq \delta_k \sqrt{1 - \frac{\delta_k^2}{4R^2}} \leq \delta_k \exp\left(-\frac{\delta_k^2}{8R^2}\right),$$

(lembrando que  $R = \max\{d(v_j, p) \mid v_j \in \mathcal{A}\}$ ) em que a última inequação segue do fato que  $1 + x \leq \exp(x)$ . Além disso, como  $\delta_k > \varepsilon R$ , segue a seguinte desigualdade

$$\exp\left(-\frac{\delta_k^2}{8R^2}\right) < \exp\left(-\frac{\varepsilon^2}{8}\right) < 1.$$

Com isso e com o Teorema 3.7 conseguimos formalizar a complexidade do Algoritmo 5 no próximo resultado.

**Teorema 3.8** ([37, Theorem 9]). *O Algoritmo do Triângulo apresenta a seguinte complexidade de iteração:*

- (a) se  $p \in \text{conv}(\mathcal{A})$ , dados  $\varepsilon > 0, p_0 \in \text{conv}(\mathcal{A})$ , com  $\delta_0 = d(p, p_0)$ , o número máximo de iterações  $K_\varepsilon$  para computar um ponto  $p_\varepsilon \in \text{conv}(\mathcal{A})$  tal que  $d(p_\varepsilon, p) < \varepsilon R$  é de

$$K_\varepsilon \leq \frac{48}{\varepsilon^2} = O(\varepsilon^{-2}). \quad (28)$$

- (b) se  $p \notin \text{conv}(\mathcal{A})$ , o número de iterações  $K_\Delta$  para computar uma  $p$ -testemunha é de

$$K_\Delta \leq \frac{48R^2}{\Delta^2} = O\left(\frac{R^2}{\Delta^2}\right),$$

em que  $\Delta := \min\{d(x, p) \mid x \in \text{conv}(\mathcal{A})\}$ .

A complexidade de iteração  $O(\varepsilon^{-2})$  no Teorema 3.8(i) implica que a taxa de convergência é apenas sublinear. No entanto, quando  $p \notin \text{conv}(\mathcal{A})$ , o Teorema 3.8(ii) mostra que a complexidade de iteração de TA depende apenas da geometria do problema

(ou seja, das constantes  $\Delta$  e  $R$ ). Isto ocorre porque o mínimo de  $d(p', p)$  para  $p' \in \text{conv}(\mathcal{A})$  será  $\Delta = \min\{d(x, p) \mid x \in \text{conv}(\mathcal{A})\}$ . Então, tomando  $\varepsilon = \frac{\Delta}{R}$  e usando (28) obtemos

$$K_{\Delta} \leq \frac{48R^2}{\Delta^2} = O\left(\frac{R^2}{\Delta^2}\right).$$

Vale ressaltar que a priori não sabemos se  $p$  pertence ou não a  $\text{conv}(\mathcal{A})$ , então o que podemos afirmar é que em

$$O\left(\max\left\{\frac{R^2}{\Delta^2}, \frac{48}{\varepsilon^2}\right\}\right)$$

iterações o TA para com uma  $\varepsilon$ -solução ou com uma testemunha de que  $p \notin \text{conv}(\mathcal{A})$ .

Com o objetivo de melhorar essa taxa de convergência quando  $p \in \text{conv}(\mathcal{A})$  uma definição mais restritiva de pivô é apresentada a seguir [37, Definição 8].

**Definição 3.9** (Pivô estrito). Sejam  $p \in \mathbb{R}^m$  e  $p_k \in \text{conv}(\mathcal{A})$ . Dizemos que  $v \in \mathcal{A}$  é um  $p$ -pivô estrito para  $p_k$ , se o ângulo  $\angle p_k p v$  entre os segmentos  $[p_k, p]$  e  $[p, v]$  é tal que  $\angle p_k p v \geq \pi/2$ .

Assim como para pivô simples, conseguimos caracterizações de pivô estrito que estão no lema a seguir.

**Lema 3.10** ([21, Lema 3.7]). Sejam  $p' \in \text{conv}(\mathcal{A})$ ,  $v_j \in \mathcal{A}$  e  $p \in \mathbb{R}^m$  o ponto a ser consultado. As seguintes afirmações são equivalentes:

- (a)  $\|v_j - p\|^2 \leq \|v_j - p'\|^2 - \|p' - p\|^2$ ;
- (b)  $2v_j^T(p' - p) \leq 2p^T(p' - p)$ ;
- (c)  $(p' - p)^T(v_j - p') \leq -\|p' - p\|^2$ ;
- (d)  $(p' - p)^T(v_j - p) \leq 0$ .

Do Lema 3.5(i) e Lema 3.10(i) fica evidente que todo pivô estrito é pivô simples, mas que a recíproca não é verdadeira, mostrando que ser pivô estrito é de fato mais restritivo. Além disso, a interpretação geométrica de um pivô estrito pode ser vista na Figura 5, em que o hiperplano  $\bar{H}$  é definido como  $\bar{H} := \{y \in \mathbb{R}^m \mid (p - p_k)^T(y - p) = 0\}$ . Analogamente a quando  $v$  é pivô simples, no caso de  $v$  ser um pivô estrito, temos

$$\delta_{k+1} = \frac{\delta_k r}{\sqrt{r^2 + \delta_k^2}} \leq \frac{\delta_k R}{\sqrt{R^2 + \delta_k^2}} \leq \delta_k \sqrt{1 - \frac{\delta_k^2}{2R^2}} \leq \delta_k \exp\left(-\frac{\delta_k^2}{4R^2}\right).$$

Destacamos que também há um teorema de dualidade de distância para pivôs estritos que é apresentado a seguir.

**Teorema 3.11** ([37, Teorema 10]). Suponha  $p \notin \mathcal{A}$ . Então, temos  $p \in \text{conv}(\mathcal{A})$  se, e somente se, para cada  $p_k \in \text{conv}(\mathcal{A})$  existe um pivô estrito  $v \in \mathcal{A}$ .

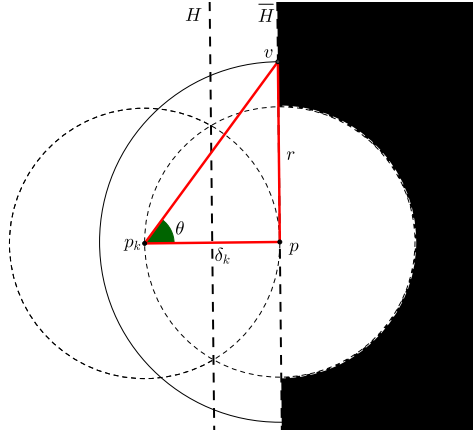


Figura 5 – A área sombreada representa a região onde os pivôs estreitos podem ser encontrados.

Apesar da inspiração para definição de um pivô estrito ser a de melhorar a complexidade quando  $p \in \text{conv}(\mathcal{A})$ , da mesma forma que o Teorema 3.8, o Algoritmo do Triângulo usando pivôs estreitos também requer  $\mathcal{O}(1/\varepsilon^2)$  iterações para encontrar uma  $\varepsilon$ -solução (embora a constante multiplicativa seja menor; veja [37, Teorema 11]).

No entanto, pivôs estreitos serão úteis para descrever uma situação em que a taxa de convergência é linear. Primeiro, observe de (27) que a redução em  $\delta_k = d(p_k, p)$ , a cada iteração do Algoritmo do Triângulo, depende do ângulo  $\theta_k = \angle pp_k v$  (veja a Figura 4). Portanto, se  $\cos \theta_k$  ficar afastado de zero, é possível melhorar os resultados de complexidade para TA. De fato, formalizamos isso, que foi estabelecido em [37], com o auxílio da próxima hipótese.

**Hipótese 3.12.** Existe uma constante  $c > 0$  (chamada de *fator de visibilidade*) tal que a seguinte desigualdade é satisfeita

$$\sin \theta' \leq \frac{1}{\sqrt{1+c}}, \quad \forall p' \in \text{conv}(\mathcal{A}) \setminus B_{\varepsilon R}(p), \quad (30)$$

em que  $\theta' = \angle pp'v$  e  $v$  é um pivô para  $p'$ .

*Observação 3.13.* Dado  $\varepsilon \in (0, 1)$ , na Seção 7 do artigo [37] é provado que para

$$\theta^* = \max\{\theta' = \angle pp'v \mid v \in \mathcal{A}, p' \in \text{conv}(\mathcal{A}), \|p - v\|^2 \leq \|p' - v\|^2, \|p' - p\| > \varepsilon R\},$$

temos que

$$\sin \theta^* \leq \frac{1}{\sqrt{1+\varepsilon^2}}.$$

Note que usando o Algoritmo do Triângulo a cada iteração temos  $\theta_k = \angle pp_k v_j$  e  $v_j$  é um pivô para  $p_k$  e pela Observação 3.13,  $\sin \theta_k \leq \sin \theta^*$ . Isto é utilizado na demonstração do teorema a seguir.

**Teorema 3.14** ([37, Theorem 13]). *Seja  $\delta_0 = d(p_0, p)$ ,  $p_0 \in \text{conv}(\mathcal{A})$  e assuma que a Hipótese 3.12 é satisfeita.*

- (a) *Se  $p \in \text{conv}(\mathcal{A})$ , então o número de iterações para o Algoritmo do Triângulo obter uma  $p_\varepsilon \in \text{conv}(\mathcal{A})$  tal que  $d(p_\varepsilon, p) \leq \varepsilon R$  é de*

$$\mathcal{O}\left(\frac{1}{c} \ln \frac{\delta_0}{\varepsilon R}\right).$$

- (b) *Se  $p \notin \text{conv}(\mathcal{A})$  o número de iterações para o Algoritmo do Triângulo obter uma  $p$ -testemunha é de*

$$\mathcal{O}\left(\frac{1}{c} \ln \frac{\delta_0}{\Delta}\right).$$

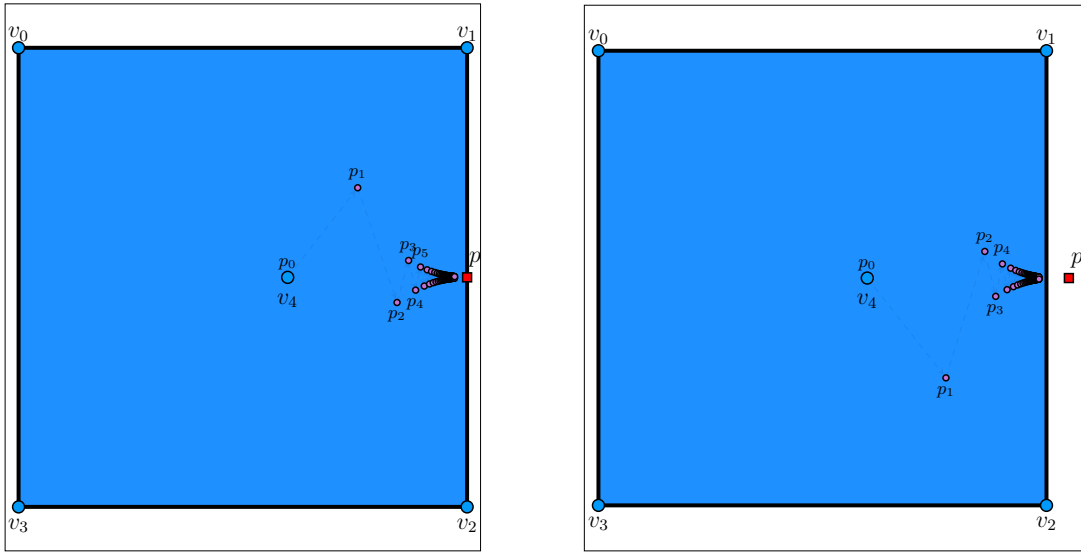
As complexidades de iteração do Teorema 3.14 são melhores em comparação com as do Teorema 3.8, desde que o fator de visibilidade  $c$  seja limitado e continue longe de zero. Como veremos adiante, este é o caso, por exemplo, quando  $p$  pertence ao interior relativo de  $\text{conv}(\mathcal{A})$ . No entanto, em outros casos,  $c$  pode estar muito próximo de 0 e, como consequência, as complexidades do Teorema 3.14 podem ser piores do que as do Teorema 3.8. Quando isso acontece, o TA pode experimentar um fenômeno de ziguezague, como mostra o Exemplo 3.15.

*Exemplo 3.15.* Considere  $\mathcal{A} := \{v_0, v_1, v_2, v_3, v_4\} \subset \mathbb{R}^2$  em que  $\{v_0, v_1, v_2, v_3\}$  são vértices de um quadrado unitário e  $v_4$  é um ponto interior. Assuma que o ponto de  $\mathcal{A}$  mais próximo de  $p$  é  $v_4$  (que foi gerado deslocando um pouco o centro do quadrado para a direita). Analisamos duas situações representadas na Figura 6. Na Figura 6a, nós temos que o ponto a ser consultado  $p$  é o ponto médio de uma das arestas do  $\text{conv}(\mathcal{A})$  e na Figura 6b mostramos o caso em que  $p$  é movido para uma posição fora do quadrado. Em ambos os casos, o comportamento em zigue-zague do TA aparece. Nós apenas mostramos as primeiras 60 iterações do TA, pois o Algoritmo 5: leva mais de um milhão de iterações para alcançar  $\|p_k - p\| \leq \varepsilon R$ , com  $\varepsilon = 10^{-4}$  no caso (a) e 81 iterações para encontrar uma testemunha no caso (b). Para  $p \in \text{conv}(\mathcal{A})$ , temos que  $\delta_0 = 0.40$  e  $R = 1.12$ . Monitoramos a sequência  $(\sin \theta_k)_{k \in \mathbb{N}}$  baseado nas iterações do TA com o objetivo de estimar um limitante superior para (30). Notamos que os valores se aproximam de 1 na medida em que  $k$  cresce<sup>1</sup>. No caso em que  $p \notin \text{conv}(\mathcal{A})$ , temos  $\delta_0 = 0.45$  e  $\Delta = 0.05$ . Novamente, observar  $(\sin \theta_k)_{k \in \mathbb{N}}$  nos permitiu fornecer um limite superior para a constante de visibilidade  $c$  de 0.0044.

Sob a hipótese adicional de  $p$  estar no interior relativo de  $\text{conv}(\mathcal{A})$ , pode-se mostrar que tal  $c > 0$  que satisfaz (30) existe. Além disso, podemos obter um limite inferior para ele, em termos da geometria do problema.

**Hipótese 3.16.**  $B_\rho(p)$  está contida no interior relativo de  $\text{conv}(\mathcal{A})$ .

<sup>1</sup> Para os pontos que calculamos, se o fator de visibilidade  $c$  existir, ele não será maior que  $10^{-7}$ .


 (a)  $p \in \text{conv}(\mathcal{A})$  na fronteira do quadrado

 (b)  $p \notin \text{conv}(\mathcal{A})$ 

Figura 6 – Quadrado Unitário do Exemplo 3.15 com as primeiras 60 iterações do TA.

Seja  $p_k$  o iterado atual tal que  $d(p, p_k) > \varepsilon R$ . Se, para um dado  $\rho > 0$ , a bola  $B_\rho(p)$  está estritamente no interior relativo de  $\text{conv}(\mathcal{A})$ , então existe um  $q \in \text{conv}(\mathcal{A})$  que pertence à extensão do segmento  $[p_k, p]$  (veja Figura 7) com  $d(q, p) = \rho$ . Note que, se existe um pivô estrito  $v$  satisfazendo  $(p - q)^T(v - q) \leq 0$ , então

$$\begin{aligned} \rho R \cos \angle qpv &\geq \|p - q\| \|v - p\| \cos \angle qpv = (q - p)^T(v - p) \\ &= (q - p)^T(v - q + q - p) = \|q - p\|^2 - (p - q)^T(v - q) \geq \rho^2 \end{aligned}$$

implicando que

$$\sin \theta_k \leq \sin \angle qpv \leq \sqrt{1 - \rho^2/R^2} \leq \frac{1}{\sqrt{1 + \rho^2/R^2}}. \quad (31)$$

Ressaltamos que (31) é necessário na prova do próximo teorema [37, Teorema 14], e este foi obtido assumindo a desigualdade  $(p - q)^T(v - q) \leq 0$ . Assim consideramos a seguinte hipótese.

**Hipótese 3.17.** Seja  $q = p + \tau(p - p_k)$  para  $\tau > 0$  tal que  $q \in \text{conv}(\mathcal{A})$  e  $d(q, p) = \rho$ . O Algoritmo do Triângulo usa, a cada iteração, um pivô estrito  $v$  que satisfaz

$$(p - q)^T(v - q) \leq 0. \quad (32)$$

**Teorema 3.18.** Sejam  $R = \max\{d(v_i, v_j) \mid v_i, v_j \in \mathcal{A}\}$ ,  $p_0 \in \text{conv}(\mathcal{A})$ ,  $\delta_0 = d(p_0, p)$ , e suponha que as Hipóteses 3.16 e 3.17 são satisfeitas. Para um dado  $\varepsilon \in (0, 1)$ , o número máximo de iterações que o Algoritmo 5 leva para computar um  $p_\varepsilon \in \text{conv}(\mathcal{A})$ , tal que  $d(p, p_\varepsilon) \leq \varepsilon R$ , é de

$$\mathcal{O}\left(\frac{R^2}{\rho^2} \ln \frac{\delta_0}{\varepsilon R}\right).$$

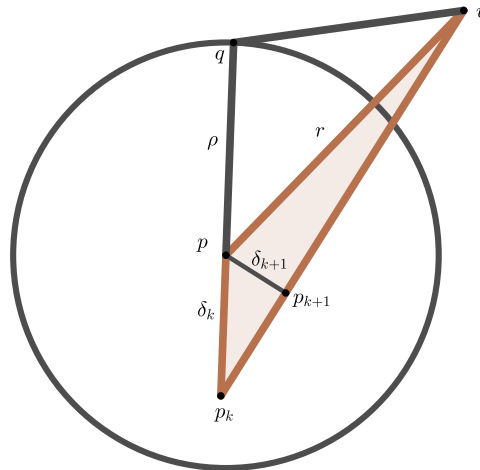


Figura 7 – Um pivô que satisfaz a condição (32)

Este resultado implica a convergência linear do Algoritmo 5, usando pivôs estritos satisfazendo a Hipótese 3.17, quando  $p \in B_\rho(p) \subset \text{relint}(\text{conv}(\mathcal{A}))$ .

*Observação 3.19.* A prova do Teorema 3.18 segue as linhas de [37, Teorema 14]. Embora o enunciado do Teorema 14 em [37] exija apenas o uso de um pivô estrito em cada iteração, a condição (32) é essencial para obter a desigualdade (31) que é usada na prova do teorema. Aqui destacamos que um pivô estrito não, necessariamente, satisfaz a condição (32). Note que um pivô estrito,  $v$ , satisfazendo (32) é representado na Figura 7. Na Figura 8, a área cinza representa a região em que podemos encontrar pivôs estritos. Observe que  $v_1, v_2$  e  $v_3$  são pivôs estritos, mas  $v_2$  não satisfaz (32) pois está no lado *errado* do hiperplano  $\bar{H}_q := \{y \in \mathbb{R}^m \mid (p - q)^T(y - q) = 0\}$ . De fato, somente pivôs estritos que estão na área hachurada satisfazem (32).

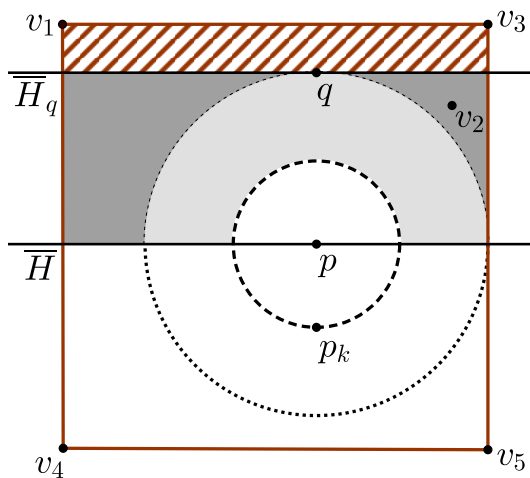


Figura 8 – Diferença entre apenas pivô estrito e um pivô estrito que satisfaça a condição (32)

Na próxima seção discutiremos em detalhes as semelhanças e diferenças do Algo-

ritmo do Triângulo com o método de Frank-Wolfe, quando aplicado ao CHMP. Além disso, mostraremos também que Frank-Wolfe aplicado a (22), sempre que possível, escolhe  $v \in \mathcal{A}$  que satisfaz as hipóteses do Teorema 3.18.

### 3.3 FRANK-WOLFE APLICADO AO CHMP

Note que se aplicarmos o algoritmo de Frank-Wolfe (Algoritmo 1) à formulação (22), usando  $y_k$  no lugar de  $x_k$  e  $\text{conv}(\mathcal{A})$  no lugar de  $C$ , em cada iteração, devemos resolver o subproblema

$$\begin{aligned} \min_y \quad & (y_k - p)^T (y - y_k) \\ \text{s. a.} \quad & y \in \mathcal{A}. \end{aligned} \quad (33)$$

Como  $\mathcal{A} = \{v_1, \dots, v_n\}$  é um conjunto finito,  $\bar{y}_k$  solução de (33) pode ser escolhido como  $\bar{y}_k = v_j$ , tal que  $v_j^T (y_k - p) = \min_{1 \leq i \leq n} v_i^T (y_k - p)$ . Levando em conta que  $y_k \in \text{conv}(\mathcal{A})$  para cada  $k$ , podemos associar  $y_k$  ao iterado  $p_k$  do Algoritmo 5 e re-escrever

$$v_j^T (p_k - p) = \min_{1 \leq i \leq n} v_i^T (p_k - p). \quad (34)$$

Em comparação, na  $k$ -ésima iteração, o Algoritmo 5 escolhe  $v_i \in \mathcal{A}$  tal que  $v_i^T (p_k - p) \leq (\|p_k\|^2 - \|p\|^2)/2$ , que, pelo Lemma 3.5, caracteriza um pivô. Assim, fica claro por (34), que o Algoritmo 1 não apenas escolhe um pivô, mas aquele que minimiza o produto interno  $v_i^T (p_k - p)$ . Nesse sentido, Frank-Wolfe (Algoritmo 1) aplicado a (22), com *busca linear exata*, pode ser visto como uma versão *gananciosa* do Algoritmo do Triângulo.

Por outro lado, o Algoritmo do Triângulo pode ser interpretado como uma versão *inexata* de Frank-Wolfe, no seguinte sentido: semelhante a [35], consideramos  $\tilde{y}_k \in C$  uma solução inexata do subproblema FW (veja o passo 2 do Algoritmo 1 trocando  $x$  por  $y$  e  $f$  por  $\Psi$ ), se a seguinte desigualdade for satisfeita

$$(\tilde{y}_k - y_k)^T \nabla \Psi(y_k) \leq \min_{y \in C} (y - y_k)^T \nabla \Psi(y_k) + \frac{1}{2} \hat{\delta}_k \bar{\gamma}_k C_\Psi,$$

em que  $\hat{\delta}_k \geq 0$  é um parâmetro de inexatidão,  $\bar{\gamma}_k$  é dado em (25) e  $C_\Psi$  é a *constante de curvatura* de  $\Psi$  em relação ao conjunto  $C$  (veja [35] para uma definição formal e discussão detalhada sobre tal constante).

Se considerarmos (22), temos  $\Psi(y) = \|y - p\|^2/2$ ,  $\nabla \Psi(y) = y - p$ , e  $C = \text{conv}(\mathcal{A})$  (para o qual  $C_\Psi = D^2$ ). Então, identificando  $y_k$  com  $p_k$  e definindo o parâmetro de inexatidão como

$$\hat{\delta}_k = \frac{2}{D^2} \|v_j - p_k\|^2, \quad (35)$$



em que  $j \in \arg \min\{v_i^T(p_k - p) \mid 1 \leq i \leq n\}$ , se  $v_i$  é um pivô, temos

$$\begin{aligned} \min_{y \in C} (y - p_k)^T \nabla f(p_k) + \frac{1}{2} \tilde{\gamma}_k C_f \hat{\delta}_k &= (v_j - p_k)^T (p_k - p) + \frac{1}{2} \tilde{\gamma}_k D^2 \frac{2}{D^2} \|v_j - p_k\|^2 \\ &= (v_j - p_k)^T (p_k - p) - \frac{(p_k - p)^T (v_j - p_k)}{\|v_j - p_k\|^2} \|v_j - p_k\|^2 \\ &= 0 \geq -\frac{1}{2} \|p_k - p\|^2 \geq (p_k - p)^T (v_i - p_k), \end{aligned} \quad (36)$$

no qual na segunda desigualdade usamos Lema 3.5(iii). Assim, (36) mostra que o pivô  $v_i$  na  $k$ -ésima iteração do Algoritmo 5, é uma solução inexata para o subproblema FW para  $\hat{\delta}_k$  como em (35).

Portanto, não é surpreendente que os resultados de complexidade para ambos os algoritmos sejam semelhantes.

Conforme discutido na Seção 2.3.1 (veja desigualdade (12)), se  $y^*$  é solução de (22), as iterações do método de Frank-Wolfe satisfazem

$$\Psi(y_k) - \Psi(y^*) = \mathcal{O}\left(\frac{1}{k}\right).$$

Se  $p \in \text{conv}(\mathcal{A})$ , dado  $\varepsilon \in (0, 1)$ , de acordo com o Teorema 3.8, o Algoritmo do Triângulo em  $K_\varepsilon \leq 48\varepsilon^{-2}$  iterações obtém um ponto  $p_\varepsilon \in \text{conv}(\mathcal{A})$  tal que  $d(p_\varepsilon, p) < \varepsilon R$ .

Dado um índice  $k$ , da desigualdade  $\lceil 48\varepsilon^{-2} \rceil \geq k$ , temos que  $\varepsilon \leq \sqrt{48/k}$ , e assim

$$\|p_k - p\| < \sqrt{\frac{48}{k}} R.$$

Equivalentemente,

$$\Psi(p_k) = \Psi(p_k) - \Psi(y^*) < \frac{24R^2}{k} = \mathcal{O}\left(\frac{1}{k}\right),$$

mostrando que TA tem a mesma taxa de convergência que Frank-Wolfe aplicado ao CHMP.

Curiosamente, a complexidade da iteração é melhorada para ambos os algoritmos quando  $p$  está no interior relativo de  $\text{conv}(\mathcal{A})$  conforme Teorema 3.18 para o Algoritmo do Triângulo com pivôs estritos e como pode ser deduzido a partir de (16) para Frank-Wolfe (veja também [31, Theorem 2]).

Na Tabela 1, nós apresentamos uma comparação entre a complexidade de iteração do Algoritmo do Triângulo, para o caso em que  $p \in \text{conv}(\mathcal{A})$ , e a dos algoritmos Frank-Wolfe e Away Step Frank-Wolfe (ASFW) sob diferentes hipóteses. A dedução das constantes “geométricas” reportadas é detalhada no Apêndice B.

Contudo, note que quando  $p \notin \text{conv}(\mathcal{A})$ , de acordo com o Teorema 3.8(ii), as iterações do Algoritmo do Triângulo dependem apenas da “geometria do problema”, isto é, não dependem da tolerância  $\varepsilon$ . Como o FW pode ser visto como um Algoritmo do Triângulo Ganancioso, essa afirmação também é válida, desde que os critérios de parada

Tabela 1 – Complexidade de iteração para encontrar  $p_k \in \text{conv}(\mathcal{A})$  tal que  $d(p_k, p) < \varepsilon R$ . A última hipótese aplica-se apenas ao TA.

Hipóteses	TA	FW	ASFW
$p \in \text{conv}(\mathcal{A})$	$\mathcal{O}(1/\varepsilon^2)$	$\mathcal{O}(D^2/\varepsilon^2)$	$\mathcal{O}\left(\frac{D^2(m+1)^2}{\Omega_C^2} \ln \frac{\delta_0}{\varepsilon}\right)$
com Hipótese 3.16	$\mathcal{O}(1/\varepsilon^2)$	$\mathcal{O}\left(\frac{D^2}{\rho^2} \ln \frac{\delta_0}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{D^2(m+1)^2}{\Omega_C^2} \ln \frac{\delta_0}{\varepsilon}\right)$
com Hipótese 3.17	$\mathcal{O}\left(\frac{R^2}{\rho^2} \ln \frac{\delta_0}{\varepsilon R}\right)$	$\mathcal{O}\left(\frac{D^2}{\rho^2} \ln \frac{\delta_0}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{D^2(m+1)^2}{\Omega_C^2} \ln \frac{\delta_0}{\varepsilon}\right)$

baseados no teorema de dualidade de distância (Teorema 3.3) sejam integrados ao FW (faremos isso na Seção 3.5).

Com respeito ao custo computacional por iteração, embora no pior caso ambos os algoritmos tenham que avaliar toda a lista de produtos internos  $v_i^T(p_k - p)$ , espera-se que, em geral, a iteração do Algoritmo 5 apresente um custo menor pois pede apenas um pivô simples, em contraste com o método de Frank-Wolfe que precisa do pivô que minimiza o produto interno  $v_i^T(p_k - p)$ .

### 3.3.1 Frank-Wolfe como um Algoritmo do Triângulo Ganancioso

Acabamos de investigar a relação entre o Algoritmo do Triângulo e o método de Frank-Wolfe aplicado ao problema (22): FW é equivalente ao Algoritmo do Triângulo quando escolhermos um pivô  $v_j$  tal que

$$v_j \in \arg \min \{v_i^T(p_k - p) \mid v_i \in \mathcal{A} \setminus \{p_k\}\}. \quad (37)$$

A partir do Lemma 3.5(ii), o subproblema (37) mostra que o método de Frank-Wolfe (com as peculiaridades do CHMP) é equivalente a um Algoritmo do Triângulo *ganancioso*, no qual, ao invés de escolher  $v_j$  tal que

$$v_j^T(p_k - p) \leq \frac{\|p_k\|^2 - \|p\|^2}{2}, \quad (38)$$

selecionamos  $v_j \in \mathcal{A} \setminus \{p_k\}$  tal que o lado esquerdo de (38) é mínimo. Chamaremos esse algoritmo de *Triângulo Ganancioso* (ou simplesmente GT, do inglês Greedy Triangle) e o sintetizamos no Algoritmo 6.

*Observação 3.20.* Ressaltamos que, para  $p = 0$ , o Algoritmo 6 coincide com um algoritmo proposto por Von Neumann<sup>2</sup> [20, 27, 51].

*Observação 3.21.* O critério de parada do Passo 7 é baseado na caracterização de pivô estrito presente no Lema 3.10.

<sup>2</sup> Comunicado por von Neumann a Dantzig na década de 1940 e estudado posteriormente por Dantzig em [18].

**Algoritmo 6:** ALGORITMO DO TRIÂNGULO GANANCIOSO

---

**Dados:**  $\mathcal{A}, p, R, \varepsilon \in (0, 1)$

- 1 Escolha  $p_0 \in \arg \min\{d(v_j, p) \mid v_j \in \mathcal{A}\}$  ;
- 2 **para**  $k = 0, 1, 2, \dots$  **faça**
- 3     **se**  $d(p_k, p) < \varepsilon R$  **então**
- 4         | pare
- 5     **fim**
- 6     Escolha  $v_j = \arg \min\{v_i^T(p_k - p) \mid v_i \in \mathcal{A} \setminus \{p_k\}\}$  .
- 7     **se**  $v_j^T(p_k - p) > p^T(p_k - p)$  **então**
- 8         | pare:  $p_k$  é uma  $p$ -testemunha.
- 9     **fim**
- 10     Seja  $\bar{\gamma}_k = -\frac{(p_k - p)^T(v_j - p_k)}{\|v_j - p_k\|^2}$ .
- 11      $p_{k+1} \leftarrow (1 - \bar{\gamma}_k)p_k + \bar{\gamma}_k v_j$ .
- 12 **fim**

---

Sabendo que a inequação  $v_j^T(p_k - p) \leq p^T(p_k - p)$  caracteriza um pivô estrito, note que escolhendo  $v_j$  como no Passo 6 do Algoritmo 6 temos que um pivô estrito é sempre escolhido. Além disso, se houver pivôs estritos verificando a condição (32) (veja Hipótese 3.17), o Algoritmo 6 escolherá um deles.

**Proposição 3.22.** *Assuma que a Hipótese 3.16 é satisfeita.*

- (i) *Se  $d(p_k, p) > \varepsilon R$ , então existe um pivô estrito  $w \in \mathcal{A}$  satisfazendo a Hipótese 3.17.*
- (ii) *O Algoritmo 6 sempre escolhe  $v_j \in \mathcal{A}$  satisfazendo a Hipótese 3.17.*

*Demonstração.* (i) Seja  $p_k \in \text{conv}(\mathcal{A})$  tal que  $d(p, p_k) > \varepsilon R$ . A Hipótese 3.16 implica que existe  $\tau > 0$  tal que  $q = p + \tau(p - p_k) \in \text{conv}(\mathcal{A})$  e  $d(q, p) = \rho$ . Como  $q \in \text{conv}(\mathcal{A})$ , aplicando o Teorema 3.11 para  $q$  (ao invés de  $p$ ), nós concluímos que existe um  $q$ -pivô estrito  $w$  em  $p$ . Então, a inequação (32) segue do Lema 3.10(iv).

- (ii) Considere  $v_j$  como no Passo 6 do Algoritmo 6 e seja  $w \in \mathcal{A}$  um pivô estrito satisfazendo a Hipótese 3.17. Então,

$$\begin{aligned}
 (p - q)^T(v_j - q) &= -\tau(p - p_k)^T(v_j - p - \tau(p - p_k)) \\
 &= -\tau(p - p_k)^T v_j + \tau(p - p_k)^T p + \tau^2 \|p - p_k\|^2 \\
 &\leq -\tau(p - p_k)^T w + \tau(p - p_k)^T p + \tau^2 \|p - p_k\|^2 \\
 &= (p - q)^T(w - q) \leq 0.
 \end{aligned}$$

Consequentemente,  $v_j$  também satisfaz a desigualdade (32).

□

Uma possível desvantagem do Algoritmo 6, em relação com o Algoritmo 5, é que, a cada iteração, é necessário percorrer toda a lista  $\mathcal{A}$  para garantir o mínimo de (37). Portanto, o custo por iteração é sempre  $\mathcal{O}(nm)$ .

### 3.4 GRADIENTE PROJETADO APLICADO AO CHMP

Uma vez que o problema (21) é um caso particular do problema (8), em que  $f \equiv \Phi$  e  $C = \Delta_n$  (com  $x \in \mathbb{R}^n$ ), podemos aplicar os métodos de Gradiente Projetado, que foram revisados na Seção 2.3.2.

Vale destacar que o principal custo por iteração dos métodos de gradiente projetado é o cálculo da projeção  $P_C(\cdot)$ . Felizmente, a projeção sobre  $C = \Delta_n$  pode ser calculada a um custo razoável de  $\mathcal{O}(n \log n)$  operações (para mais detalhes, consulte [16]).

É fundamental destacar que no caso em que  $p \notin \text{conv}(\mathcal{A})$ , o Algoritmo do Triângulo para sua iteração ao encontrar uma testemunha. No entanto, o método de Gradiente Projetado aplicado ao problema (21) irá parar próximo da projeção de  $p$  no  $\text{conv}(\mathcal{A})$ . Por este motivo é importante incorporar ao método de gradiente projetado critérios de parada específicos para o CHMP (o mesmo acontece para métodos do tipo Frank-Wolfe aplicados ao problema (22)). Tais critérios serão discutidos na próxima seção.

### 3.5 CRITÉRIOS DE PARADA ESPECÍFICOS PARA CHMP

A dualidade de distância (Teorema 3.3) é um resultado notável para o CHMP, e é a chave do Algoritmo do Triângulo. Assim, é natural indagar se outros métodos de primeira ordem para resolver o CHMP podem se beneficiar de tal resultado.

A partir da discussão da Seção 3.3.1, estabelecemos que Frank-Wolfe aplicado ao CHMP pode ser visto como um Algoritmo do Triângulo Ganancioso, no sentido de escolher  $v_j \in \mathcal{A} \setminus \{p_k\}$  que minimiza  $v_j^T(p_k - p)$ . Dessa forma, no caso que  $v_j^T(p_k - p) > p^T(p_k - p)$ , podemos afirmar que não há pivô estrito (pela caracterização vista no Lema 3.10) a partir de  $p_k$  e  $p \notin \text{conv}(\mathcal{A})$ . Assim, como no Algoritmo do Triângulo, é possível parar o FW (e suas variantes) assim que uma testemunha for detectada. Além disso, como no TA, também paramos as iterações de FW assim que  $\|p_k - p\| \leq \varepsilon R$ .

Outra questão importante é como definir a tolerância  $\epsilon$  do Algoritmo 1 (Passo 4) para garantir  $\|p_k - p\| \leq \varepsilon R$ . Observe que o critério de parada clássico do Algoritmo 1 (e do Algoritmo 2) é  $\nabla \Psi(y_k)^T(\bar{y}_k - y_k) \geq -\epsilon$  (aqui denotamos a iteração por  $y_k$  ao invés de  $x_k$  pois estamos considerando o problema (22)). Seja  $y^* \in \text{conv}(\mathcal{A})$  a solução de (22). Da Seção 2.3.1.1, para o problema (22) temos que

$$\Psi(y_k) - \Psi(y^*) \leq \nabla \Psi(y_k)^T(y_k - \bar{y}_k) \leq \epsilon,$$

em que  $\bar{y}_k$  é solução do subproblema de FW. Isso é equivalente a

$$\|y_k - p\|^2 - \|y^* - p\|^2 \leq 2\epsilon,$$

implicando que  $\|y_k - p\| - \|y^* - p\| \leq 2\epsilon / \|y_k - p\|$ . Sendo assim, impondo  $\epsilon := \|y_k - p\| \varepsilon R / 2$ , temos

$$\|y_k - p\| \leq \|y^* - p\| + \varepsilon R. \quad (39)$$

Se  $p \in \text{conv}(\mathcal{A})$ ,  $y^*$ , solução de (22), é tal que  $\|y^* - p\| = 0$  e então a inequação (39) coincide com  $\|p_k - p\| \leq \varepsilon R$  (para  $p_k = y_k$ ).

Por outro lado, se  $\nabla \Psi(y_k)^T(y_k - \bar{y}_k) \leq \|y_k - p\| \varepsilon R/2$  e  $\|y_k - p\| > \varepsilon R$  então, de (39),  $\|y^* - p\| > 0$  e concluímos que  $p \notin \text{conv}(\mathcal{A})$ . Portanto, em vez do critério de parada clássico para FW, podemos usar o critério de *erro relativo*

$$\nabla \Psi(y_k)^T(y_k - \bar{y}_k) \leq \|y_k - p\| \frac{\varepsilon R}{2}.$$

Quando se trata de métodos de gradiente projetado aplicados ao problema (21), lembramos que  $\nabla \Phi(x_k) = A^T(Ax_k - p)$  e, desde que  $x_k \in \Delta_n$ , podemos escrever  $p_k = Ax_k \in \text{conv}(\mathcal{A})$ . Assim, cada componente do gradiente pode ser escrito como  $[\nabla \Phi(x_k)]_i = v_i^T(p_k - p)$  e, caso cada componente seja maior que  $p^T(p_k - p)$ , podemos parar retornando  $p_k = Ax_k$  como uma testemunha de que  $p \notin \text{conv}(\mathcal{A})$ .

Em seguida, discutimos como definir a tolerância  $\varepsilon$  no Algoritmo 4 (Passo 4) para garantir que  $\|A\bar{x}_k - p\| \leq \varepsilon R$  (em que  $\bar{x}_k$  é do Passo 2). Para isso, primeiro enunciamos e provamos o seguinte lema.

**Lema 3.23.** *Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função convexa com gradiente  $L$ -Lipschitz e  $C \subset \mathbb{R}^n$  um conjunto não vazio, convexo e compacto, com diâmetro  $D$ . Se  $x^* \in C$  é o minimizador de  $f$  sobre  $C$  e  $\bar{x} := P_C(x - \lambda \nabla f(x))$ , para algum  $\lambda > 0$  e  $x \in C$ , então*

$$f(\bar{x}) - f(x^*) \leq D \left( \frac{L}{2} + \frac{1}{\lambda} \right) \|\bar{x} - x\|.$$

*Demonstração.* Como  $\bar{x}$  é a projeção de  $x - \lambda \nabla f(x)$  sobre  $C$ , pela caracterização de projeções em conjuntos convexos (veja Teorema 2.9) temos  $(w - \bar{x})^T(x - \lambda \nabla f(x) - \bar{x}) \leq 0$ , para todo  $w \in C$ . Em particular, como  $x^* \in C$  e  $\lambda > 0$ , ao fazer  $w = x^*$ , obtemos

$$\nabla f(x)^T(\bar{x} - x^*) \leq \frac{1}{\lambda}(\bar{x} - x^*)^T(x - \bar{x}). \quad (40)$$

Note também que

$$\begin{aligned} (\bar{x} - x^*)^T(x - \bar{x}) &= (\bar{x} - x)^T(x - \bar{x}) + (x - x^*)^T(x - \bar{x}) \\ &\leq -\|x - \bar{x}\|^2 + \|x - x^*\| \|x - \bar{x}\|. \end{aligned} \quad (41)$$

Portanto, usando (40) e (41), conseguimos, após algumas manipulações,

$$\nabla f(x)^T(\bar{x} - x) \leq \nabla f(x)^T(x^* - x) + \frac{1}{\lambda}(-\|x - \bar{x}\|^2 + \|x - x^*\| \|x - \bar{x}\|). \quad (42)$$

Como  $\nabla f$  é  $L$ -Lipschitz, usando (42) e a convexidade de  $f$ , temos que

$$f(\bar{x}) - f(x) - \frac{L}{2} \|x - \bar{x}\|^2 \leq f(x^*) - f(x) + \frac{1}{\lambda}(-\|x - \bar{x}\|^2 + \|x - x^*\| \|x - \bar{x}\|),$$

implicando em

$$f(\bar{x}) - f(x^*) \leq \frac{L}{2} \|x - \bar{x}\|^2 + \frac{1}{\lambda}(-\|x - \bar{x}\|^2 + \|x - x^*\| \|x - \bar{x}\|) \leq D \left( \frac{L}{2} + \frac{1}{\lambda} \right) \|\bar{x} - x\|,$$

em que, na última inequação, usamos o fato de  $x, x^* \in C$ , que tem, por hipótese, diâmetro  $D$ .  $\square$

Agora, lembre que para o problema (21),  $f(x) := \Phi(x) = \frac{1}{2} \|Ax - p\|^2$  e  $C = \Delta_n$ . Nesse caso,  $L = \|A\|^2$  e  $D = \sqrt{2}$ . Da definição de parâmetro espectral (veja (19)), temos que  $1/\lambda_k \leq L$ . Então, podemos reescrever a desigualdade do Lema 3.23 como

$$\Phi(\bar{x}_k) - \Phi(x^*) \leq \frac{3}{2} LD \|\bar{x}_k - x_k\|, \quad (43)$$

tomando  $x_k$  como  $x$  e  $\bar{x}_k$  como  $\bar{x}$ .

Como no Algoritmo 4,  $d_k := \bar{x}_k - x_k$ , e o critério de parada usual é  $\|d_k\| < \epsilon$ , se impomos  $\epsilon := \frac{\|A\bar{x}_k - p\|}{3LD} \epsilon R$ , a inequação (43) implica que  $\|A\bar{x}_k - p\| \leq \|Ax^* - p\| + \epsilon R$ . Se  $p \in \text{conv}(\mathcal{A})$ ,  $x^*$ , solução de (21), é tal que  $\|Ax^* - p\| = 0$  e então a última desigualdade coincide com  $\|A\bar{x}_k - p\| \leq \epsilon R$ . Observamos que o cálculo de  $\bar{x}_k$  tal que a última desigualdade seja satisfeita é semelhante a encontrar uma  $\epsilon$ -solução para o Algoritmo do Triângulo já que  $A\bar{x}_k \in \text{conv}(\mathcal{A})$ . Por outro lado, se  $\|d_k\| \leq \frac{\|A\bar{x}_k - p\|}{3LD} \epsilon R$  e  $\|A\bar{x}_k - p\| > \epsilon R$ , então  $\|Ax^* - p\| > 0$  e podemos concluir que  $p \notin \text{conv}(\mathcal{A})$ .

Com estes resultados, conseguimos aproveitar a dualidade de distâncias (Teorema 3.3) em outros métodos de primeira ordem para resolver o CHMP, possibilitando que estes interrompam as iterações ao detectar que não há pivô (estrito) a partir do iterado atual. Além disso, os critérios de parada usuais de Frank-Wolfe e Gradiente Projetado também foram substituídos por novos critérios de parada usando “erro relativo” que permitem detectar uma  $\epsilon$ -solução para o CHMP, ou que  $p \notin \text{conv}(\mathcal{A})$ . Na próxima seção iremos apresentar diversos experimentos numéricos envolvendo o Algoritmo do Triângulo, sua versão Gananciosa, Away-Step Frank-Wolfe e Gradiente Projetado Espectral (os dois últimos já com os novos critérios de parada desta seção).

### 3.6 EXPERIMENTOS NUMÉRICOS

Para avaliar o desempenho dos métodos de primeira ordem descritos nas Seções 3.2, 3.3, 3.4 munidos dos critérios de parada específicos (Seção 3.5) para resolução do CHMP, apresentamos três conjuntos de experimentos computacionais.

Primeiramente, na Seção 3.6.1, consideramos exemplares do CHMP gerados artificialmente, organizados em 4 casos diferentes de acordo com a distribuição dos pontos de  $\mathcal{A}$  e a posição relativa do ponto de consulta  $p$ . Dessa forma temos algum controle sobre os valores das constantes  $R$ ,  $\delta_0$ ,  $\Delta$  e  $c$  (dos Teoremas 3.8 e 3.14), e podemos avaliar o desempenho dos algoritmos em condições favoráveis e desfavoráveis. Nas duas seções seguintes, estudamos duas potenciais aplicações do CHMP. Na Seção 3.6.2, consideramos problemas de factibilidade de programação linear que, sob hipóteses adequadas, podem ser expressos como CHMPs. Na Seção 3.6.3 aplicamos os métodos estudados para um problema de classificação de imagem, a fim de determinar se elementos do conjunto de teste pertencem ao envoltório convexo dos elementos no conjunto de treinamento.

Todos os testes foram executados em computador pessoal com processador Intel Core i7-8565U 1.80 GHz com 8 GB de RAM rodando Windows 10. Os algoritmos foram

implementados em Matlab R2019a. Os códigos de nossos experimentos estão disponíveis em <https://github.com/rafaelafilipposzi/First-order-methods-for-the-CHMP>.

### 3.6.1 Exemplos Artificiais do CHMP

Os exemplos artificiais do CHMP foram gerados seguindo um procedimento semelhante ao proposto por [47]. Cada elemento de  $\mathcal{A} = \{v_1, \dots, v_n\}$  é gerado segundo uma distribuição uniforme na bola unitária em  $\mathbb{R}^m$  [32]. Para isto, cada um dos  $n$  pontos de  $\mathcal{A}$  é gerado da seguinte maneira: primeiro geramos um vetor  $\hat{v} \in \mathbb{R}^m$  com componentes amostradas independentemente de uma distribuição normal padrão; então definimos  $v \in \mathcal{A}$  como  $v := \frac{\hat{v}}{\|\hat{v}\|}$ , em que  $u$  é amostrado da distribuição uniforme em  $[0, 1]$ .

O ponto a ser consultado  $p \in \mathbb{R}^m$  é gerado nos quatro casos a seguir:

- (a)  $p \in \text{conv}(\mathcal{A})$  no interior relativo de  $\text{conv}(\mathcal{A})$ ;
- (b)  $p \in \text{conv}(\mathcal{A})$  com fator de visibilidade próximo de zero;
- (c)  $p \notin \text{conv}(\mathcal{A})$  longe da fronteira;
- (d)  $p \notin \text{conv}(\mathcal{A})$  com fator de visibilidade próximo de zero.

Os casos acima foram selecionados com o objetivo de avaliar o desempenho dos algoritmos de primeira ordem nos melhores e piores casos para o Algoritmo do Triângulo. Para o caso (a) temos a complexidade de iteração melhorada dada pelo Teorema 3.18 para o Algoritmo do Triângulo com “pivôs fortes” (veja Hipóteses 3.16, 3.17 e Proposição 3.22). Nos casos em que o ponto  $p$  está fora de  $\text{conv}(\mathcal{A})$ , a posição relativa de  $p$  afeta os valores de  $\Delta$ ,  $R$  e  $\delta_0$  nos Teoremas 3.8 e 3.14. Nos casos (b) e (d) tentamos forçar o fator de visibilidade  $c$  a ser próximo de zero para obter exemplos mais difíceis para os algoritmos do Triângulo (TA e GT). Os detalhes são fornecidos nas Seções 3.6.1.2 e 3.6.1.4.

Para todos os testes nesta seção, consideramos as seguintes tolerâncias, parâmetros algorítmicos e detalhes de implementação:

- O número máximo de iterações foi definido como  $\min\{\max\{10^3 n, 10^5\}, 10^6\}$ .
- Os critérios de parada usuais de Away Step Frank-Wolfe (ASFW) e Spectral Projected Gradient (SPG) foram alterados de acordo com as discussões na Seção 3.5.
- A tolerância para encontrar uma  $\varepsilon$ -solução para o CHMP foi definida em  $\varepsilon := 10^{-4}$  para todos os algoritmos testados.
- Para a busca linear não monótona do SPG foi utilizado  $M = 15$  pois obtive o melhor desempenho em testes preliminares. Além disso, definimos  $\lambda_{\min} := 10^{-8}$  e  $\lambda_{\max} := 10^8$ .
- Como as variáveis  $x \in \mathbb{R}^n$  na formulação quadrática (21) são os coeficientes da combinação convexa (dos elementos de  $\mathcal{A}$ ), e não o ponto  $p_k \in \mathbb{R}^m$  como no Algoritmo do Triângulo, consideramos o ponto inicial  $x_0 = e_i \in \mathbb{R}^m$ , em que  $e_i$

denota o  $i$ -ésimo vetor canônico de  $\mathbb{R}^n$  e o índice  $i$  é tal que

$$i = \arg \min_{1 \leq j \leq n} \{\|v_j - p\|\},$$

assim  $Ax_0 = p_0$ , em que  $p_0$  é o ponto inicial para o Algoritmo 5.

- Na nossa implementação do Algoritmo do Triângulo, a cada iteração, construímos a lista completa de valores  $v_i^T(p_k - p)$  (veja Lema 3.5(ii)), encontramos todos os pivôs (se houver) e selecionamos um aleatoriamente. A escolha aleatória de um pivô (em vez de um pivô  $v_i$  com o menor índice  $i$ ) é inspirada por [57] e acaba sendo muito eficaz na redução do número de iterações de TA nos quatro casos considerados em nossos experimentos. Embora pudéssemos testar sequencialmente se  $v_i$ , para  $i = 1, 2, \dots, n$ , é um pivô e avaliar os produtos  $v_i^T(p_k - p)$  um por um, parando a busca assim que o primeiro pivô é encontrado, nós “vetorizamos” a busca, o que normalmente tem um desempenho superior no Matlab. Enfatizamos que no TA usamos pivôs simples (veja Lema 3.5) em cada iteração ao invés de pivôs estritos como na Definição 3.9.

O custo computacional por iteração do TA e do GT é  $\mathcal{O}(mn)$  operações aritméticas, correspondendo a  $n$  produtos internos de vetores em  $\mathbb{R}^m$ . No ASFW, também temos operações aritméticas  $\mathcal{O}(mn)$  por iteração para determinar as direções Frank-Wolfe e Away. SPG é o algoritmo que tem o maior custo computacional por iteração,  $\mathcal{O}(2mn + n \log n)$ , pois requer dois produtos matriz-vetor e a projeção no simplex unitário  $\Delta_n$ ; veja Seção 3.4.

*Observação 3.24.* Os trabalhos [36, 39] mostram que com um pré-processamento de  $\mathcal{O}(mn^2)$ , o custo de cada iteração do Algoritmo do Triângulo pode ser reduzido para  $\mathcal{O}(m + n)$ . Essa estratégia pode ser interessante quando o número estimado de iterações é muito maior que o número de pontos em  $\mathcal{A}$ . No entanto, destacamos que tal estratégia não foi utilizada em nossos experimentos.

Nos experimentos reportados nessa subseção, fixamos a dimensão como  $m = 100$  e variamos o número de pontos  $n$  em  $\mathcal{A}$  de 500 a  $10^5$ . Para cada valor de  $n$ , foram gerados 10 exemplares aleatórios. As tabelas no Apêndice C mostram o número *médio* de iterações exigidas por cada algoritmo.

### 3.6.1.1 caso (a): $p$ no interior relativo de $\text{conv}(\mathcal{A})$

Para garantir que o ponto de consulta esteja no interior relativo do  $\text{conv}(\mathcal{A})$ , selecionamos  $p = 0 \in \mathbb{R}^m$ , o centro da bola unitária. Como cada  $v_j \in \mathcal{A}$  é originado a partir de uma distribuição uniforme na bola unitária, sempre que o número  $n$  de pontos em  $\mathcal{A}$  for suficientemente grande, com alta probabilidade temos que  $p \in \text{conv}(\mathcal{A})$ . Além do mais, também é provável que  $B_\rho(p) \subset \text{conv}(\mathcal{A})$ , para  $\rho > 0$  afastado de zero e, portanto, neste caso, a Hipótese 3.16 é satisfeita.



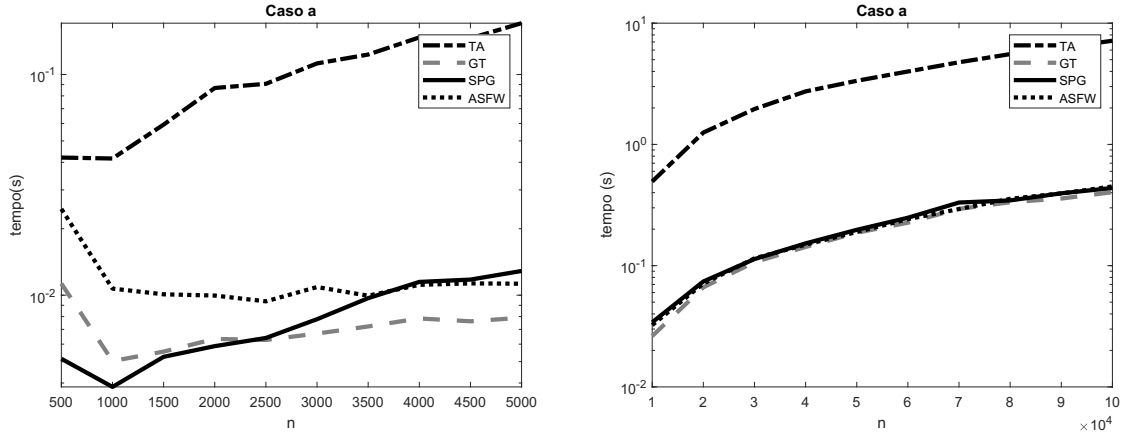


Figura 9 – Caso (a). Tempos de execução (em segundos) para a dimensão  $m = 100$  e aumentando  $n$ , o número de pontos em  $\mathcal{A}$ .

Nesse caso, esperamos que GT tenha uma performance melhor que o TA. Lembremos, dos Teoremas 3.8 e 3.14, que a complexidade por iteração do TA é

$$\mathcal{O}\left(\min\left\{\frac{1}{\varepsilon^2}, \frac{1}{c} \ln \frac{\delta_0}{\varepsilon R}\right\}\right),$$

em que  $\delta_0 = d(p, p_0)$ ,  $R := \max\{d(v_i, p) \mid v_i \in \mathcal{A}\}$  e  $c$  é da Hipótese 3.12. No entanto, tendo em conta a Proposição 3.22, a Hipótese 3.17 é satisfeita pelo GT e sua complexidade de iteração é dada pelo Teorema 3.18

$$\mathcal{O}\left(\frac{R^2}{\rho^2} \ln \frac{\delta_0}{\varepsilon R}\right).$$

A Figura 9 mostra o desempenho dos algoritmos em termos de tempo de execução, enquanto o número de iterações está listado na Tabela 6 (no Apêndice C). Conforme previsto, GT requer menos iterações do que TA para recuperar  $p_\varepsilon$  tal que  $d(p_\varepsilon, p) \leq \varepsilon R$ . Mais ainda, quando  $n$  é grande, GT também alcança o melhor desempenho em termos de tempo de execução, seguido de perto por SPG e ASFW. Embora o custo por iteração do SPG seja maior, ele exigiu apenas 12 iterações em média, enquanto GT e ASFW usaram até 118 iterações.

### 3.6.1.2 Caso (b): $p \in \text{conv}(\mathcal{A})$ com fator de visibilidade perto de zero

Aqui, nós geramos um ponto  $p \in \text{conv}(\mathcal{A})$  na fronteira do envoltório convexo, semelhante ao Exemplo 3.15. Para isso, primeiro encontramos os dois pontos de  $\mathcal{A}$ ,  $v_\ell$  e  $v_q$ , que têm os maiores valores do funcional linear  $\psi(v) = e^T v$ : como  $\text{conv}(\mathcal{A})$  é um polítopo e pela forma como os pontos de  $\mathcal{A}$  foram aleatoriamente gerados, esperamos, para um  $n$  suficientemente grande, que  $v_\ell$  e  $v_q$  sejam pontos extremos de  $\text{conv}(\mathcal{A})$  (o máximo de  $\psi(v)$

é alcançado em pelo menos um desses pontos). Em seguida, definimos  $p = v_\ell/2 + v_q/2$ . Além disso, para evitar que  $v_\ell$  e  $v_q$  sejam pontos de  $\mathcal{A}$  mais próximos de  $p$ , adicionamos a  $\mathcal{A}$  um novo ponto  $v_s$  tal que  $d(v_s, p) < d(v_\ell, v_q)/2$ . Mais precisamente,

$$v_s = p - \frac{\beta \|v_\ell - v_q\|}{2 \|p\|} p,$$

em que  $\beta = 0,9$ . Este último passo é necessário porque se  $p_0 = v_q = \operatorname{argmin}\{d(v, p) \mid v \in \mathcal{A}\}$ , por exemplo,  $v_\ell$  é um pivô estrito uma vez que

$$(p_0 - p)^T(v_q - p) = (v_\ell - p)^T(v_q - p) = \frac{1}{4}(v_\ell - v_q)^T(v_q - v_\ell) \leq 0.$$

Nesta situação, GT poderia selecionar  $v_\ell$  como pivô na primeira iteração e  $p$  seria recuperado após a busca linear exata. De fato isso aconteceu quando não adicionamos  $v_s$ .

Os exemplares considerados nesta subseção são mais difíceis para algoritmos TA e GT porque o limite superior  $1/\sqrt{1+c}$  em (30) pode ser muito próximo de 1. Como  $p = \frac{1}{2}(v_q + v_\ell)$ ,  $v_q$  ou  $v_\ell$  é um pivô para qualquer  $p' \in \operatorname{conv}(\mathcal{A}) \setminus \{p\}$ . De fato, caso contrário

$$(p' - p)^T(v_\ell - p) > \frac{1}{2} \|p' - p\|^2 \quad \text{e} \quad (p' - p)^T(v_q - p) > \frac{1}{2} \|p' - p\|^2$$

implicaria em

$$0 = (p' - p)^T(v_\ell + v_q - 2p) > \|p' - p\|^2,$$

uma contradição. Além disso, se  $v_\ell$  foi o pivô escolhido para obter  $p_k$  na iteração  $k-1$ , da busca linear exata  $(v_\ell - p_k)^T(p - p_k) = 0$ , portanto  $v_\ell$  não é um pivô na iteração  $k$ , o que implica que  $v_q$  é. Portanto, começando em  $p_0 \in \operatorname{conv}(\mathcal{A}) \setminus \{v_\ell, v_q, p\}$ , poderíamos ter em cada iteração de TA (ou GT)  $p_k$  como uma combinação convexa de  $p_0$ ,  $v_\ell$  e  $v_q$ , com  $v_q$  e  $v_\ell$  alternando como pivôs. Este fenômeno é semelhante ao do Exemplo 3.15. Conseqüentemente, já que  $1/c$  pode ser relativamente grande, esperamos observar a complexidade do Teorema 3.8 em vez da do Teorema 3.14 para TA e GT (em contraste com o caso (a)). No entanto, as hipóteses do Teorema 2.24, para ASFW, ainda são satisfeitas.

Os algoritmos do Triângulo e do Triângulo Ganancioso atingiram o número máximo de iterações e, portanto, não são relatados na Figura 10 e na Tabela 6. Em média, o ASFW convergiu em 12 iterações, e em 7 delas a direção “away” foi usada. Da Figura 10, vemos que o ASFW superou o SPG. Isso ocorreu porque, embora o SPG tenha atingido a precisão desejada em no máximo 10 iterações (em média veja Tabela 6), este possui um custo por iteração maior que o ASFW.

### 3.6.1.3 Caso (c): $p \notin \operatorname{conv}(\mathcal{A})$ e afastado da fronteira

Agora, o ponto a ser consultado  $p$  está localizado fora de  $\operatorname{conv}(\mathcal{A})$ . Geramos  $p \in \mathbb{R}^m$  como no caso (b), seguido de uma dilatação: multiplicamos por 1,5. Diferentemente do

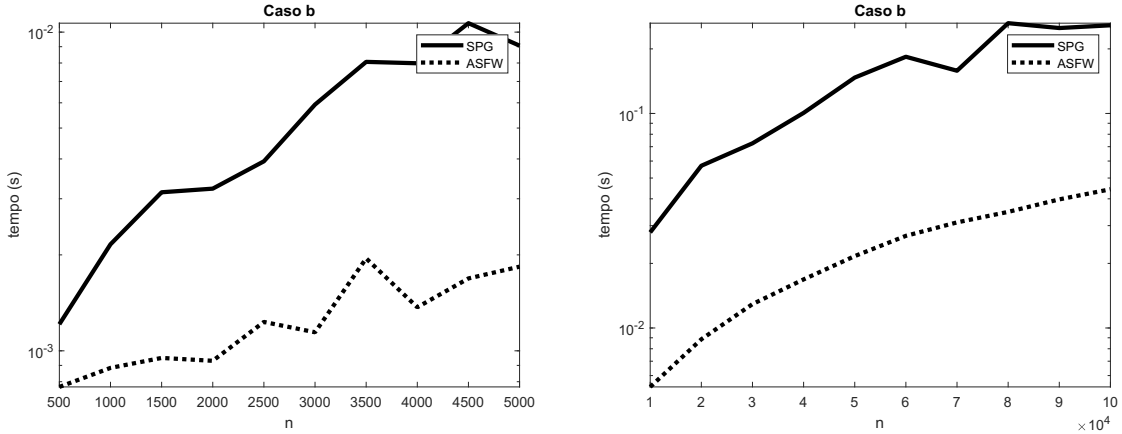


Figura 10 – Caso (b). Tempos de execução (em segundos) para a dimensão  $m = 100$  e aumentando  $n$ , o número de pontos em  $\mathcal{A}$ .

caso anterior, não adicionamos outro ponto  $v_s$  a  $\text{conv}(\mathcal{A})$  para evitar que  $p_0$  seja  $v_q$  ou  $v_\ell$ . Observe que, dessa forma, como  $\mathcal{A} \subset B_1(0)$ ,  $p \notin \text{conv}(\mathcal{A})$  e  $p$  está suficientemente distante da fronteira. Isso garante que  $\Delta := \min\{d(v, p) \mid v \in \text{conv}(\mathcal{A})\}$  seja suficientemente maior que zero. Como pode ser visto na Tabela 7 (Apêndice C), todos os algoritmos encontraram uma testemunha em não mais que 4 iterações. Vale lembrar que o Teorema 3.8 afirma que a complexidade de iteração, neste caso, não depende da tolerância  $\varepsilon$ , mas das constantes  $\Delta$  e  $R$  (não é difícil mostrar que a razão  $R/\Delta$  se aproxima de 1 à medida que  $\Delta$  aumenta). A Figura 11 mostra os tempos de execução para este caso. Lembramos dos Teoremas 3.8 e 3.14 que a complexidade de iteração de TA e GT, quando  $p \notin \text{conv}(\mathcal{A})$ , é

$$\mathcal{O}\left(\min\left\{\frac{R^2}{\Delta^2}, \frac{1}{c} \ln \frac{\delta_0}{\Delta}\right\}\right),$$

em que  $c$  vem de (30) e depende da escolha do pivô. Como GT procura um pivô  $v$  que minimiza  $v^T(p_k - p)$ , normalmente  $\sin \theta_k$  será menor para GT do que TA (veja Figura 5) e, portanto, esperamos uma constante  $c$  maior para GT do que TA, o que explica o menor número de iterações para o primeiro.

Para os exemplares considerados nesta subseção, observamos  $1,58 \leq R \leq 1,8$ ,  $0,32 \leq \Delta \leq 0,41$  e  $0,68 \leq \delta_0 \leq 0,81$ . Além disso, avaliando  $\sin \theta_k$  ao longo das iterações, deduzimos que se o fator de visibilidade  $c$  existe, seu valor deve ser menor que 0,02.

Além disso, o ASFW e o GT têm um desempenho melhor do que o SPG. A diferença no tempo de execução entre ASFW (e GT) e SPG aumenta à medida que o número de pontos aumenta. Embora o número de iterações exigidas pelo SPG seja semelhante aos demais, lembramos que seu custo por iteração é maior devido ao custo de projeção no simplex unitário ( $\mathcal{O}(n \log n)$ ).

Observamos que nestes exemplares  $p_0$ , o ponto de  $\mathcal{A}$  mais próximo de  $p$ , pode ser

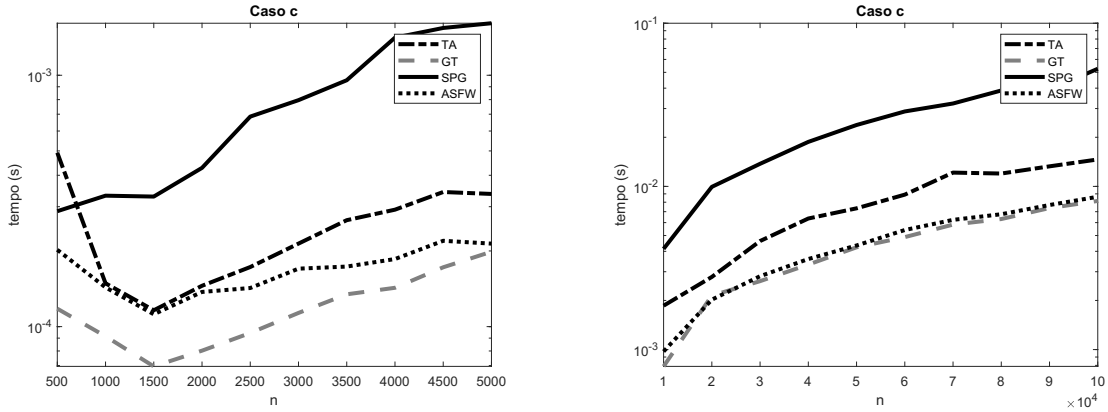


Figura 11 – Caso (c). Tempos de execução (em segundos) para a dimensão  $m = 100$  e aumentando  $n$ , o número de pontos em  $\mathcal{A}$ .

$v_q$  ou  $v_\ell$ . Conforme discutido na seção anterior, se  $p_0 = v_q$ , então  $v_\ell$  é um pivô estrito e pode ser escolhido por GT e ASFW na primeira iteração. Como, para tais exemplares, a projeção de  $p$  no  $\text{conv}(\mathcal{A})$  é uma combinação convexa de  $v_q$  e  $v_\ell$ , esses dois algoritmos podem encontrá-la na primeira iteração, explicando os resultados na Tabela 7.

#### 3.6.1.4 Caso (d): $p \notin \text{conv}(\mathcal{A})$ com fator de visibilidade perto de zero.

Motivado novamente pelo Exemplo 3.15, agora geramos  $p$  como no caso (b) mas multiplicamos suas coordenadas por 1.01. Aqui adicionamos  $v_s$ , como no caso (b) e, conforme explicado na Seção 3.6.1.2, esperamos obter exemplares com fator de visibilidade próximo de zero. De fato, analisando as iterações do TA para este conjunto de exemplares, observamos que se existir um fator positivo  $c$  seu valor será menor que  $\mathcal{O}(10^{-4})$ . Além disso, para os exemplares gerados neste caso, temos  $0.007 \leq \Delta \leq 0,008$ ,  $1,34 \leq R \leq 1,54$  e  $0,57 \leq \delta_0 \leq 0,75$ . Assim, a partir do Teorema 3.8 (e do Teorema 3.14) no pior caso TA pode gastar  $\mathcal{O}(10^4)$  iterações para encontrar uma testemunha.

Como pode ser visto na Tabela 7, o número médio de iterações de TA e GT é muito alto em comparação com os outros algoritmos e isso se traduz nos tempos de execução mostrados na Figura 12. Observamos que o número de iterações do SPG e ASFW também aumentou neste caso quando comparado ao caso (c). Observe que o Algoritmo do Triângulo leva praticamente o mesmo tempo que o GT.

Notavelmente, as direções “away” permitiram ao ASFW evitar o fenômeno de zigue-zague visto no Exemplo 3.15: sendo o algoritmo mais rápido, seguido pelo SPG.

### 3.6.2 Problema de Factibilidade de Programação Linear

O problema de inclusão no envoltório convexo também está relacionado ao problema de factibilidade de programação linear. O conjunto factível de um problema de

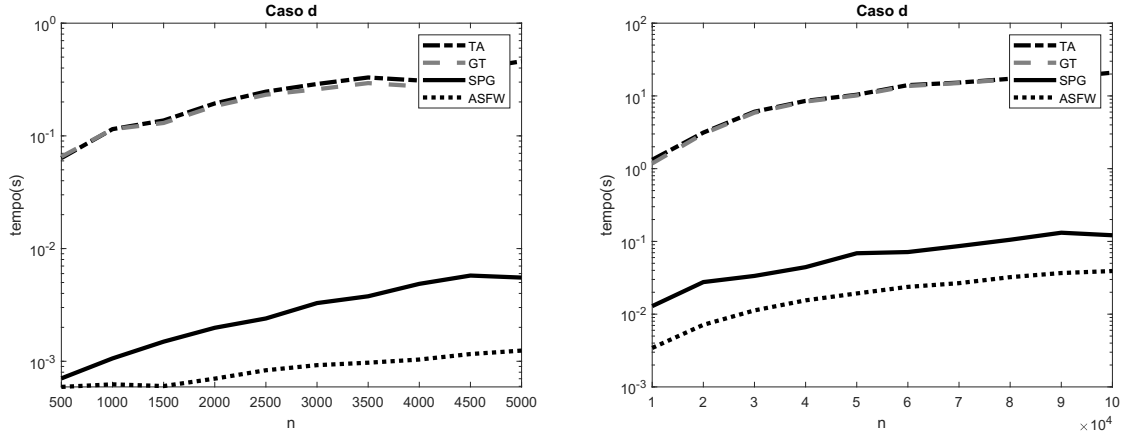


Figura 12 – Caso (d). Tempos de execução (em segundos) para a dimensão  $m = 100$  e aumentando  $n$ , o número de pontos em  $\mathcal{A}$ .

programação linear pode ser dado por  $\Omega = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ , com matriz  $A = [a_1 \ a_2 \ \dots \ a_n]$ , e vetores  $b, a_i \in \mathbb{R}^m$ . Suponha que estamos interessados em elementos de  $\Omega$  com norma limitada, digamos  $\|x\|_1 \leq N$ . Em outras palavras, gostaríamos de decidir se a interseção de  $\Omega$  com o semi-espaço definido por  $H_N = \{x \in \mathbb{R}^n \mid e^T x \leq N\}$ , para um dado  $N > 0$ , não é vazia. Segundo [42], este (mais específico) problema de factibilidade de PL pode ser formulado como

$$\begin{pmatrix} A & 0 & -b \\ e^T & 1 & -N \\ 0^T & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{N+1} \end{pmatrix}, \quad (44)$$

$$e^T \alpha + \beta + \gamma = 1,$$

$$\alpha, \beta, \gamma \geq 0,$$

que é equivalente ao CHMP para  $\text{conv}(\tilde{\mathcal{A}})$ , em que o ponto a ser consultado é  $p = \left(0^T, 0, \frac{1}{N+1}\right)^T \in \mathbb{R}^{m+2}$  e os elementos de  $\tilde{\mathcal{A}}$  são as colunas da matriz de coeficientes em (44). Não é difícil mostrar que  $\Omega \cap H_N$  não é vazio se e somente se  $p \in \text{conv}(\tilde{\mathcal{A}})$ .

*Observação 3.25.* Seja  $p_k \in \text{conv}(\tilde{\mathcal{A}})$  tal que  $d(p_k, p) = \|p_k - p\| \leq \varepsilon R$ , em que  $R$  é o diâmetro de  $\text{conv}(\tilde{\mathcal{A}})$ . Então,  $\hat{x} := \left(\frac{\alpha_1}{\gamma}, \dots, \frac{\alpha_n}{\gamma}\right)^T$  satisfaz  $\|A\hat{x} - b\| \leq \varepsilon R / \gamma =: \varepsilon'$ ,  $\left|e^T \hat{x} + \beta / \gamma - N\right| \leq \varepsilon'$ ,  $\left|\gamma - \frac{1}{N+1}\right| \leq \varepsilon R$  e  $\hat{x} \geq 0$ .

Vamos considerar exemplares do problema de factibilidade em PL

$$Ax = b, \quad x \geq 0 \quad \text{e} \quad e^T x \leq N, \quad (45)$$

e resolve-os como CHMPs usando TA, GT, SPG e ASFW, juntamente com os critérios de parada descritos na Seção 3.5, e comparar o desempenho com o solver `linprog` (que

é uma implementação do algoritmo Dual-Simplex) do Matlab aplicado a (45). Para cada exemplar, as colunas de  $A$  foram geradas da distribuição uniforme na esfera unitária centrada em  $e \in \mathbb{R}^m$  (logo  $a_{ij} \geq 0$ ). Para gerar exemplares factíveis de (45), selecionamos um vetor aleatório  $x \in \mathbb{R}^n$ , em que cada entrada é gerada da distribuição uniforme em  $(0, 1)$  e calculamos  $b$  como  $b = Ax$ . Note que a soma de  $n$  variáveis aleatórias independentes com distribuição uniforme tem valor esperado  $n/2$  e variância  $n/12$ . Nestes experimentos consideramos  $n \leq 2000$ , e para garantir  $e^T x \leq N$ , definimos  $N = 1200$ . Para exemplares não factíveis, usamos o mesmo procedimento, mas no final multiplicamos a primeira coordenada de  $b$  por  $-1$ .

Para `linprog`, testamos a factibilidade de (45) considerando uma tolerância de violação da restrição de  $10^{-3}$  (o maior valor permitido por `linprog`) e também desabilitamos os procedimentos de pré-processamento. Para os outros algoritmos aplicados à formulação CHMP equivalente, usamos os mesmos parâmetros dos experimentos da Seção 3.6.1 exceto para a tolerância  $\varepsilon$ , que aqui assumimos valores de  $\{10^{-6}, 10^{-7}\}$  e o número máximo de iterações, ficou definido como  $10^6$ . Na implementação do SPG, com base em experimentos preliminares, definimos  $\lambda_{\min} = 10^{-10}$  e  $\lambda_{\max} = 10^{10}$  e usamos  $M = 60$  para a busca linear não monótona. Vale ressaltar que, para os exemplares que geramos, a constante  $R$ , que depende da geometria do problema, fica em torno de  $10^3$ .

Para cada par  $m$  e  $n$  consideramos 10 exemplares aleatórios e as médias dos tempos de execução e os respectivos números de iterações são descritos nas Tabelas 2, 3 e 4. Na Tabela 2, relatamos o tempo e as iterações de TA, GT, ASFW e SPG para problemas factíveis, com dimensões variadas e tolerância  $\varepsilon$ . A Tabela 3 mostra o desempenho de `linprog` (para o qual a tolerância de violação de restrição é fixa).

Observe que mesmo GT e ASFW, que apresentaram os melhores desempenhos nos experimentos da Seção 3.6.1, tiveram dificuldades nesses experimentos, exigindo muitas iterações e conseqüentemente apresentando tempos de execução maiores. Isso era de alguma forma esperado porque o Teorema 3.8 declara a complexidade de iteração do TA e suas variantes como  $O(\varepsilon^{-2})$  e agora  $\varepsilon$  é três ordens de magnitude menor do que o da a seção anterior.

Quanto ao ASFW, observamos que a taxa de convergência linear que aparece na equação (18) depende de  $\mu$ ,  $L$ ,  $D$ ,  $\Omega_C$  e da dimensão do problema. Nesses experimentos,  $C := \text{conv}(\tilde{\mathcal{A}})$  e a dimensão é  $m + 2$ . A função objetivo de (22) implica que  $\mu = L = 1$ . De (44), deduzimos que  $D \geq N + 1$  e  $\Omega_C \leq 1$ . Portanto,

$$\frac{\Omega_C}{D(m+3)} \leq \frac{1}{(N+1)(m+3)}.$$

Como, em nossos experimentos,  $N = 1200$  e  $m = 10^2$  a razão acima é menor que  $10^{-5}$  e substituindo-a em (18) vemos que o fator de convergência é muito próximo de 1, justificando a lenta convergência do ASFW para esses exemplares.

A Tabela 4 mostra os resultados (em média) para problemas não factíveis. Obser-

Tabela 2 – O tempo de CPU (em segundos) e o número de iterações para problemas factíveis. Os melhores resultados estão em negrito.

		TA		GT		ASFW		SPG		
	$m$	$n$	tempo	iter	tempo	iter	tempo	iter	tempo	iter
$\varepsilon = 10^{-6}$	50	200	0.5137	86491.4	0.3272	55952.9	0.1507	6286.7	<b>0.0053</b>	56.8
	50	2000	0.2669	10318	0.0809	4472.4	0.0869	2190.3	0.2971	54.4
	100	500	1.0028	74248.7	0.5599	46952.8	1.0131	27998.3	<b>0.0083</b>	122
	200	2000	1.5370	29734.1	0.8193	17676.6	1.2437	18007.5	<b>0.1833</b>	268.5
$\varepsilon = 10^{-7}$	50	200	2.1938	330132	1.2349	189164.6	0.7452	26773.2	<b>0.0080</b>	58.5
	50	2000	0.6298	24588.7	0.1503	8173.1	0.2172	5847.9	0.5040	55.9
	100	500	4.0366	313218.6	2.1494	186011.7	4.0959	112368.9	<b>0.0148</b>	125.2
	200	2000	13.7279	245975.6	1.8809	39395.9	8.8099	123920.4	<b>0.1923</b>	266.2

Tabela 3 – Tempo (em segundos) para o linprog

$m$	$n$	factíveis	não factíveis
50	200	0.0162	0.0103
50	2000	<b>0.0294</b>	0.0189
100	500	0.0297	0.0128
200	2000	0.2180	0.0424

Tabela 4 – O tempo de CPU (em segundos) e o número de iterações para problemas não factíveis. Os melhores resultados estão em negrito.

		TA		GT		ASFW		SPG	
$m$	$n$	tempo	iter	tempo	iter	tempo	iter	tempo	iter
50	200	0.0011	21.1	<b>0.0003</b>	13	0.0084	253.9	0.0018	3.5
50	2000	0.0016	35	<b>0.0007</b>	33.2	0.0015	33.2	0.0352	11.3
100	500	0.0014	56.9	<b>0.0009</b>	45.4	0.0015	45.4	0.0053	3.7
200	2000	0.0053	109.2	<b>0.0045</b>	106.8	0.0060	106.8	0.0513	7.6

vamos que em problemas não factíveis, exceto `linprog`, os algoritmos param quando uma testemunha é encontrada e isso depende apenas da geometria do problema, não da tolerância  $\varepsilon$ . Para os exemplares consideradas nesta seção observamos que  $R$  está em torno de  $10^3$  e  $0, 14 \leq \Delta \leq 0, 99$ . Podemos ver que o algoritmo com melhor desempenho em relação ao tempo em casos factíveis foi o SPG, seguido por `linprog`. Já para problemas não factíveis, o GT encontrou mais rapidamente uma testemunha do que os outros algoritmos testados.

### 3.6.3 Um exemplo de CHMP em classificação de imagens

O banco de dados MNIST é um grande banco de dados de dígitos manuscritos, que é muito popular para validar algoritmos de classificação de imagens [46]. Ele contém 60.000 amostras de imagens em tons de cinza  $28 \times 28$  em seu conjunto de treinamento e 10.000 imagens em seu conjunto de teste.

No interessante estudo de [54] foi mostrado que, para o conjunto de dados MNIST, todos os pontos de teste estão consideravelmente fora do envoltório convexo do conjunto

de pontos de treinamento. Além disso, nesse caso, o autor sugeriu o seguinte procedimento de classificação: dado um ponto de teste, estimamos sua distância ao envoltório convexo de cada classe<sup>3</sup> e classificamos de acordo com a menor das distâncias. Os trabalhos [55] e [54] relataram uma precisão de 98,5% usando este procedimento.

Inspirados por esses estudos, aplicamos os algoritmos discutidos neste capítulo para obter uma estimativa da distância de um ponto de teste ao envoltório convexo das subclasses do conjunto de treinamento. A diferença para o trabalho de [54] é que não resolvemos o problema (21) exatamente, mas paramos os algoritmos assim que uma testemunha é encontrada. Lembramos de (24) que a distância de um ponto de teste a uma testemunha não é maior que o dobro da distância deste ponto de teste ao envoltório convexo. Assim, nos perguntamos como o uso dessa estimativa impactaria os resultados/conclusões de [54].

Primeiramente, consideramos  $\mathcal{A} \subset \mathbb{R}^{784}$  como o conjunto de treinamento (com 60.000 imagens vetorizadas). Em seguida, para cada um dos 10.000 pontos de teste, resolvemos os CHMPs correspondentes com os mesmos algoritmos<sup>4</sup> testados na Seção 3.6.1. Como em [54], descobrimos que todos os elementos do conjunto de teste não pertencem ao  $\text{conv}(\mathcal{A})$ . Da mesma forma que na Subseção 3.6.1.3, os algoritmos mais rápidos para obter uma testemunha para os 10.000 pontos de teste foram GT e ASFW, que levaram aproximadamente um tempo total de 12 e 13 minutos, respectivamente. O mais lento foi o SPG, que levou 114 minutos.

Para obter a distância exata de um ponto de teste ao envoltório convexo também consideramos uma versão clássica do SPG (sem os critérios de parada da Seção 3.5), que chamamos de PROJ. O PROJ levou um tempo total de quase 9 horas para encontrar a projeção de todos os pontos de teste no  $\text{conv}(\mathcal{A})$ .

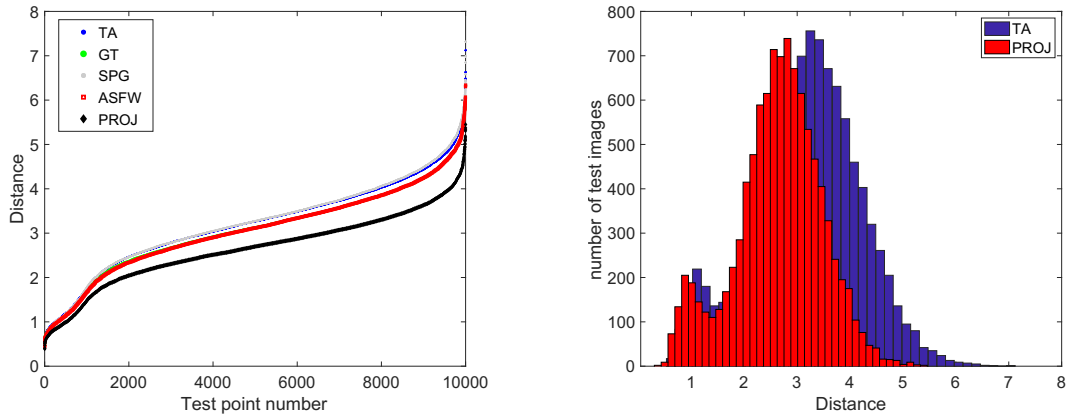
Para comparar as estimativas de distância com as distâncias exatas para o envoltório convexo do conjunto de treinamento, apresentamos na Figura 13 um gráfico com a distância (dividida por 255 que é o maior valor de pixel) de todos os pontos de teste para suas respectivas testemunhas (dependendo do método usado) e um histograma para comparar as distribuições de distância (estimativas) para TA e PROJ. Como esperado, as distâncias entre as testemunhas e os pontos de teste não ultrapassam o dobro da distância até a projeção.

Em segundo lugar, aplicamos o mesmo procedimento de classificação proposto em [55], porém calculamos a distância dos pontos de teste à testemunha no envoltório convexo de cada classe, e classificamos o ponto de teste de acordo com a classe cuja a distância da testemunha ao envoltório dos pontos de treinamento daquela classe foi a menor. A melhor precisão de 98% é alcançada pelo TA em menos de 10 minutos, o que não está muito longe da precisão de 98,5% do PROJ, que exigiu mais de 4 horas para processar todos os

<sup>3</sup> Para MNIST, as classes são  $0, 1, \dots, 9$ .

<sup>4</sup> Aqui,  $M = 3$  é usado na busca de linear não monótona para o SPG.





- (a) Estimativas de distância comparadas com a distância ao casco convexo (PROJ). (b) Histograma com as distâncias entre a testemunha e o ponto de teste (TA), e as distâncias entre o ponto de teste e sua projeção (PROJ).

Figura 13 – Resultados experimentos com MNIST.

pontos de teste.

### 3.7 CONCLUSÕES DO CAPÍTULO

Neste capítulo mostramos que é possível aplicar métodos de primeira ordem, como métodos do tipo Frank-Wolfe ou Gradiente Projetado, para o problema de inclusão no envoltório convexo e obter um desempenho competitivo em relação a algoritmos específicos para tal problema.

Para tanto, exploramos um teorema de dualidade de distância que é a base teórica para o Algoritmo do Triângulo, e desenvolvemos critérios de parada adequados para os métodos de gradiente projetado, Frank-Wolfe e Away Step Frank-Wolfe.

Os experimentos numéricos deste capítulo mostraram que estes métodos tiveram um bom desempenho, principalmente o ASFW para os casos mais desafiadores como os das Seções 3.6.1.2 e 3.6.1.4. Contudo, o mesmo não ocorreu para os problemas da Seção 3.6.2, nos quais o SPG apresentou um melhor desempenho (no geral) para exemplares factíveis, enquanto o Algoritmo do Triângulo Ganancioso mostrou-se mais eficaz para detectar não factibilidade. Por fim, para o experimentos de classificação de imagens notamos que a utilização da testemunha no lugar da projeção não altera consideravelmente a precisão da classificação, e permite grande economia em termos de tempo computacional.

## 4 CHMP ESFÉRICO E UMA VARIANTE $\mathcal{O}(1/\varepsilon)$ DO ALGORITMO DO TRIÂNGULO

Nesse capítulo buscamos melhorar a complexidade de iteração do Algoritmo do Triângulo para uma versão equivalente do CHMP: o CHMP esférico. Na Seção 4.1 apresentamos o CHMP esférico, bem como resultados envolvendo o Algoritmo do Triângulo específico para o mesmo. Veremos que quando uma propriedade específica é satisfeita pelos iterados do algoritmo, sua complexidade pode ser melhorada de  $\mathcal{O}(1/\varepsilon^2)$  para  $\mathcal{O}(1/\varepsilon)$ . Na Seção 4.2 apresentamos uma heurística que busca favorecer a ocorrência de tal propriedade. Ela envolve a resolução de um subproblema não-linear. Uma estratégia para resolução aproximada deste subproblema é apresentada na Seção 4.3, e indicam a efetividade da heurística proposta.

### 4.1 CHMP ESFÉRICO

Sejam  $\mathcal{S} := \{s_1, \dots, s_n\} \subset \mathbb{R}^m$ , com  $\|s_i\| = 1$ . O problema de inclusão no envoltório convexo *esférico* (ou CHMP-esférico) consiste em decidir se  $0 \in \text{conv}(\mathcal{S})$ .

Perceba que a partir de um CHMP com  $\mathcal{A} = \{v_1, \dots, v_n\} \subset \mathbb{R}^m$ ,  $p \in \mathbb{R}^m \setminus \mathcal{A}$  e tomando  $s_i := (v_i - p) / \|v_i - p\|$  temos um CHMP esférico. Sendo assim podemos dizer que o CHMP esférico é um caso particular do CHMP com  $p = 0$ , e com cada  $s_i \in \mathcal{S}$  tendo norma unitária. Na Figura 14 temos um exemplo para comparar o CHMP para  $\mathcal{A}$  com o CHMP-esférico relacionado. Em ambas imagens o ponto a ser consultado,  $p$  e  $0$ , respectivamente, pertence aos devidos envoltórios convexos. No trabalho de Kalantari e Zhang [42] a equivalência entre CHMP e CHMP-esférico é estabelecida. Revisaremos a seguir tal resultado e as particularidades dessa equivalência.

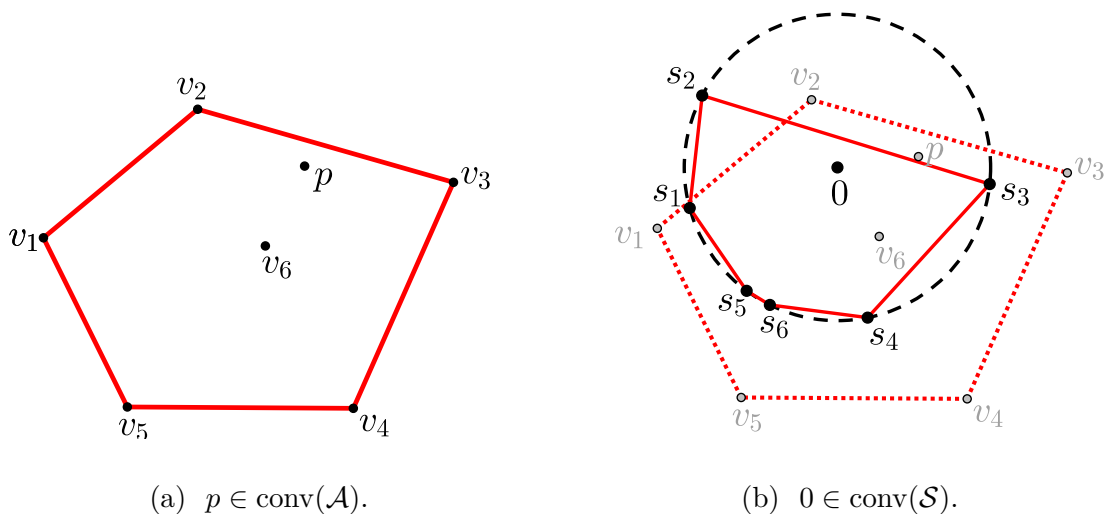


Figura 14 – Comparação do CHMP com o CHMP-esférico.

**Teorema 4.1.** *Seja  $\mathcal{A} = \{v_1, \dots, v_n\} \subset \mathbb{R}^m$ ,  $p \in \mathbb{R}^m$ ,  $p \notin \mathcal{A}$ . Defina*

$$R := \max \{\|v_i - p\| \mid v_i \in \mathcal{A}\} \text{ e } \mathcal{S} := \{s_1, \dots, s_n\} \subset \mathbb{R}^m, \text{ com } s_i := (v_i - p) / \|v_i - p\|.$$

Então,

$$p \in \text{conv}(\mathcal{A}) \text{ se, e somente se, } 0 \in \text{conv}(\mathcal{S}).$$

*Demonstração.* Suponha que  $p \in \text{conv}(\mathcal{A})$ , assim  $p = \sum_{i=1}^n \alpha_i v_i$ ,  $\sum_{i=1}^n \alpha_i = 1$ ,  $\alpha_i \geq 0$ . Logo

$$0 = \sum_{i=1}^n \alpha_i v_i - p = \sum_{i=1}^n \alpha_i v_i - \sum_{i=1}^n \alpha_i p = \sum_{i=1}^n \alpha_i (v_i - p). \quad (46)$$

Como  $p \neq v_i$ ,  $v_i - p \neq 0$  e de (46) temos

$$0 = \sum_{i=1}^n \alpha_i \|v_i - p\| \frac{(v_i - p)}{\|v_i - p\|}.$$

Dividindo ambos os lados por  $\sum_{j=1}^n \alpha_j \|v_j - p\|$  temos que  $0 \in \text{conv}(\mathcal{S})$ , pois  $0 = \sum_{i=1}^n \beta_i s_i$ ,  $\sum_{i=1}^n \beta_i = 1$  e  $\beta_i \geq 0$ , em que  $\beta_i = \frac{\alpha_i \|v_i - p\|}{\sum_{j=1}^n \alpha_j \|v_j - p\|}$ . A recíproca segue de forma análoga.  $\square$

**Teorema 4.2.** *Considere as definições do Teorema 4.1.*

(i) Dado  $\varepsilon \in (0, 1)$ , suponha que

$$\hat{p}_\varepsilon := \sum_{i=1}^n \alpha_i s_i, \text{ com } \sum_{i=1}^n \alpha_i = 1 \text{ e } \alpha_i \geq 0,$$

satisfaz  $\|\hat{p}_\varepsilon\| \leq \varepsilon$ . Então,

$$p_\varepsilon := \sum_{i=1}^n \beta_i v_i, \text{ com } \beta_i := \frac{\alpha_i / \|v_i - p\|}{\sum_{i=1}^n (\alpha_i / \|v_i - p\|)}, \quad (47)$$

satisfaz

$$\|p - p_\varepsilon\| \leq \varepsilon R.$$

(ii) Assuma  $0 \notin \text{conv}(\mathcal{S})$ . Seja  $H_{\mathcal{S}}$  o hiperplano que separa  $0$  do  $\text{conv}(\mathcal{S})$ . Então, existe um hiperplano  $H$ , paralelo ao  $H_{\mathcal{S}}$ , que separa  $p$  do  $\text{conv}(\mathcal{A})$ .

*Demonstração.* (i) Uma vez que  $\|\hat{p}_\varepsilon\| \leq \varepsilon$ , multiplicando por  $R$  temos

$$\|\hat{p}_\varepsilon\| R = \left\| \sum_{i=1}^n \alpha_i s_i R \right\| = \left\| \sum_{i=1}^n \alpha_i \frac{v_i - p}{\|v_i - p\|} R \right\| \leq \varepsilon R. \quad (48)$$

Dividindo (48) por  $\sum_{j=1}^n \frac{\alpha_j}{\|v_j - p\|} R$  e utilizando (47) concluímos

$$\left\| \sum_{i=1}^n \beta_i (v_i - p) \right\| \leq \frac{\varepsilon R}{\sum_{j=1}^n \frac{\alpha_j}{\|v_j - p\|} R}. \quad (49)$$

Pela definição de  $R$ , temos que  $\frac{1}{\|v_j - p\|} R \geq 1$ , logo

$$\sum_{j=1}^n \frac{\alpha_j}{\|v_j - p\|} R \geq \sum_{j=1}^n \alpha_j = 1.$$

Desse fato e de (49) podemos deduzir

$$\left\| \sum_{i=1}^n \beta_i p - \sum_{i=1}^n \beta_i v_i \right\| = \|p - p_\varepsilon\| \leq \varepsilon R,$$

como enunciado.

- (ii) Note que, do Teorema 4.1,  $0 \notin \text{conv}(\mathcal{S})$  se, e somente se,  $p \notin \text{conv}(\mathcal{A})$ . Vejamos agora como o hiperplano que separa  $p$  do  $\text{conv}(\mathcal{A})$  e o hiperplano que separa  $0$  do  $\text{conv}(\mathcal{S})$  estão relacionados. Sem perda de generalidade, assuma  $p = 0$ , visto que caso contrário podemos transladar  $p$  e os pontos de  $\mathcal{A}$  para termos um problema de inclusão no envoltório convexo equivalente. Sendo assim,  $R := \max\{\|v_i - p\| \mid v_i \in \mathcal{A}\} = \max\{\|v_i\| \mid v_i \in \mathcal{A}\}$ . Seja  $\mathcal{A}_R := \{v_i/R : i = 1, \dots, n\}$ . Suponha que  $p' \in \text{conv}(\mathcal{S})$  é uma testemunha de que  $0 \notin \text{conv}(\mathcal{S})$ , logo o hiperplano  $H_{\mathcal{S}} := \{x \in \mathbb{R}^m \mid p'^T x = \frac{1}{2} \|p'\|^2\}$  separa  $0$  do  $\text{conv}(\mathcal{S})$ . Assim, o semi-espaço

$$W := \left\{ x \in \mathbb{R}^m \mid p'^T x \geq \frac{1}{2} \|p'\|^2 \right\},$$

contém todo  $\mathcal{S}$ . Caso  $H_{\mathcal{S}}$  separe  $0$  do  $\text{conv}(\mathcal{A}_R)$  o teorema está demonstrado para  $H = H_{\mathcal{S}}$ . Caso contrário, existem elementos de  $\mathcal{A}_R$  que não pertencem à  $W$ . Considere  $w_i$  como a projeção de  $\frac{v_i}{R}$  sobre o segmento  $[0, p']$ , para todo  $\frac{v_i}{R} \notin W$ . Denote como  $\hat{w}$  o  $w_i$  que está mais próximo da origem, como ilustrado na Figura 15.

Para  $v_i/R \in W$ , temos

$$p'^T \frac{v_i}{R} \geq \frac{1}{2} \|p'\|^2 \geq \frac{1}{2} \|w_i\|^2 \geq \frac{1}{2} \|\hat{w}\|^2.$$

Por outro lado, para  $v_i/R \notin W$ , temos que  $p'^T(v_i/R - w_i) = 0$ , que nos leva a

$$p'^T \frac{v_i}{R} = p'^T w_i = \|p'\| \|w_i\| \geq \|\hat{w}\|^2 > \frac{1}{2} \|\hat{w}\|^2.$$

Desse modo, temos que o hiperplano  $H_R := \{x \in \mathbb{R}^m \mid p'^T x = \frac{1}{2} \|\hat{w}\|^2\}$ , separa  $0$  de  $\text{conv}(\mathcal{A}_R)$ . Equivalentemente, uma versão escalada de  $H_R$ ,  $H_R^e := \{x \in \mathbb{R}^m \mid p'^T x = \frac{R}{2} \|\hat{w}\|^2\}$  separa  $0$  do  $\text{conv}(\mathcal{A})$ .

Para  $p \neq 0$ , uma versão transladada de  $H_R^e$ , denominada  $H$ , é que separa  $p$  de  $\text{conv}(\mathcal{A})$  e temos o teorema.

□

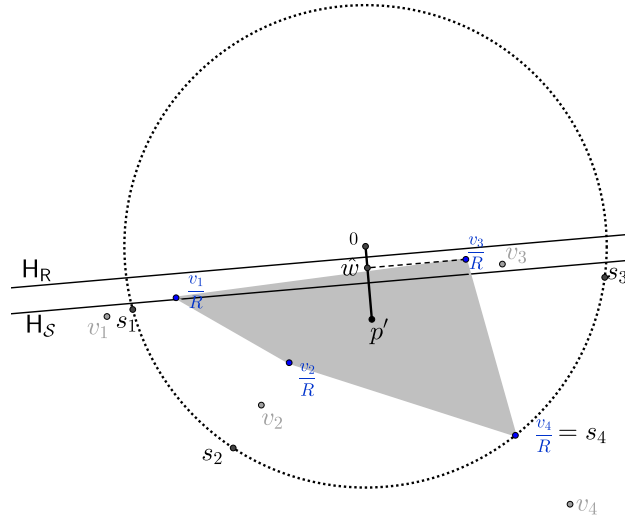


Figura 15 – Relação entre o Hiperplano que separa 0 do  $\text{conv}(\mathcal{S})$  e aquele que separa 0 do  $\text{conv}(\mathcal{A}_R)$ .

*Observação 4.3.* Além da equivalência entre o CHMP e o CHMP-esférico (Teorema 4.1), na prática é importante verificar as associações sobre as respostas de cada um desses problemas. No item (i) do Teorema 4.2 vemos as associações entre a tolerância para  $p \in \text{conv}(\mathcal{A})$  com a tolerância para aceitar  $0 \in \text{conv}(\mathcal{S})$ . Já no caso do ponto consultado estar fora do envoltório convexo, o item (ii) do Teorema 4.2 nos mostra a relação entre os hiperplanos de separação que separam  $p$  de  $\text{conv}(\mathcal{A})$  e 0 do  $\text{conv}(\mathcal{S})$ .

Uma variante natural do Algoritmo do Triângulo para resolver o CHMP-esférico é descrita no Algoritmo 7. Tal variante é denominada Algoritmo do Triângulo Esférico (ou simplesmente TA-Esférico). Note que, diferentemente de TA, a cada iteração o algoritmo procura um pivô estrito (veja Definição 3.9), além disso, o critério de parada do Passo 6 é baseado no Lema 3.10 para o caso específico do CHMP-Esférico.

A complexidade de iteração do TA-Esférico, para problemas gerais, é de  $\mathcal{O}(1/\varepsilon^2)$ , a mesma que a do TA original. No entanto, em [42] é apresentada uma complexidade de iteração melhorada quando a cada iteração o par iterado/pivô satisfaz a seguinte propriedade.

**Definição 4.4.** Considerando o CHMP-esférico,  $\varepsilon \in (0, 1)$ ,  $\frac{1}{\varepsilon} \geq M \geq 1$ . Dizemos que um ponto  $p' \in \text{conv}(\mathcal{S})$ , que não é uma testemunha e tal que  $\|p'\| > \varepsilon$ , satisfaz a  $\frac{\varepsilon}{M}$ -propriedade se existe um pivô  $s$  a partir de  $p'$  tal que

$$\|p' - s\| \geq \sqrt{1 + \frac{\varepsilon}{M}}. \quad (50)$$

O teorema a seguir apresenta uma complexidade melhorada para o TA-Esférico, quando a cada iteração o par  $(p_k, s_j)$  satisfaz a  $\frac{\varepsilon}{M}$ -propriedade.

**Algoritmo 7:** ALGORITMO DO TRIÂNGULO ESFÉRICO (TA-ESFÉRICO)

---

**Data:**  $\mathcal{S}, \varepsilon \in (0, 1)$

- 1 Escolha  $p_0 = s_1$
- 2 **para**  $k = 0, 1, 2, \dots$  **faça**
- 3     **se**  $\|p_k\| < \varepsilon$  **então**
- 4         | pare
- 5     **fim**
- 6     **se**  $s_j^T p_k > 0, \forall j = 1, \dots, n$  **então**
- 7         | pare:  $p_k$  é testemunha para  $0 \notin \text{conv}(\mathcal{S})$ .
- 8     **senão**
- 9         | Escolha  $s_j \neq p_k$ , tal que  $s_j^T p_k \leq 0$ .
- 10     **fim**
- 11      $p_{k+1} \leftarrow p_k + \frac{p_k^T (s_j - p_k)}{\|p_k - s_j\|^2} (p_k - s_j)$ .
- 12 **fim**

---

**Teorema 4.5** ([42, Theorem 9]). *Considere o TA-Esférico e suponha que  $0 \in \text{conv}(\mathcal{S})$ . Se a cada iteração o par  $p_k$  (que não é uma testemunha),  $s_j$  satisfaz a  $\varepsilon/M$ -propriedade, então em  $\mathcal{O}(M/\varepsilon)$  iterações, o TA-Esférico encontra  $p_\varepsilon \in \text{conv}(\mathcal{S})$  tal que  $\|p_\varepsilon\| \leq \varepsilon$ .*

*Demonstração.* Do Teorema 3.8 temos que se  $0 \in \text{conv}(\mathcal{S})$ , em  $\mathcal{O}(\frac{1}{\varepsilon^2})$  iterações conseguimos  $p_\varepsilon$  tal que  $\|p_\varepsilon\| \leq \varepsilon$ . Nessa lógica, temos que em  $\mathcal{O}(\frac{M}{\varepsilon})$  iterações conseguimos  $p_{k_{M\varepsilon}}$  tal que

$$\|p_{k_{M\varepsilon}}\| \leq \sqrt{\frac{\varepsilon}{M}}. \quad (51)$$

Se  $\|p_{k_{M\varepsilon}}\| \leq \varepsilon$ , a prova está terminada. Caso contrário, seja  $k := k_{M\varepsilon}$ . Então do Algoritmo 7, Passo 11, sabemos que o próximo iterado  $p_{k+1}$  é tal que

$$\|p_{k+1}\|^2 = \|p_k\|^2 - \|p_k - p_{k+1}\|^2. \quad (52)$$

Por hipótese, o pivô escolhido no passo 9 é tal que  $\|p_k - s\| \geq \sqrt{1 + \frac{\varepsilon}{M}}$ . Considere  $q$  um ponto no segmento  $[p_k, s_j]$  tal que  $\|s_j - q\| = 1$  (veja Figura 16 para uma ilustração). Note que, por definição de  $p_{k+1}$  o ângulo  $\angle 0p_{k+1}s_j$  é reto e, como  $\|s_j\| = 1$ , temos que  $\|p_k - p_{k+1}\| \geq \|p_k - q\|$  e assim

$$\|p_k - p_{k+1}\| \geq \|p_k - s_j + s_j - q\| \geq \|p_k - s_j\| - \|s_j - q\| \geq \sqrt{1 + \frac{\varepsilon}{M}} - 1. \quad (53)$$

Multiplicando e dividindo, o lado direito de (53) por  $\sqrt{1 + \frac{\varepsilon}{M}} + 1$  e lembrando que  $M \geq 1$  e  $\varepsilon \in (0, 1)$  temos que

$$\|p_k - p_{k+1}\| \geq \frac{\varepsilon}{M} \frac{1}{\sqrt{1 + \frac{\varepsilon}{M}} + 1} \geq \frac{\varepsilon}{M} \frac{1}{\sqrt{2} + 1} \geq \frac{\varepsilon}{3M}. \quad (54)$$

De (51), (52) e (54) temos que

$$\|p_{k+1}\|^2 \leq \frac{\varepsilon}{M} - \frac{\varepsilon^2}{9M^2}.$$

Com isso, após  $T$  iterações temos que

$$\|p_{k+T}\|^2 \leq \frac{\varepsilon}{M} - T \frac{\varepsilon^2}{9M^2}. \quad (55)$$

Como  $\frac{1}{\varepsilon} \geq M \geq 1$  o lado direito de (55) é não-negativo e para que  $\|p_{k+T}\| \leq \varepsilon$  é necessário que

$$T \geq \left( \frac{\varepsilon}{M} - \varepsilon^2 \right) \frac{9M^2}{\varepsilon^2} = \frac{9M}{\varepsilon} - 9M^2.$$

Assim o número de iterações  $T$  para conseguirmos  $\|p_{k+T}\|^2 \leq \varepsilon^2$  é  $\mathcal{O}(M/\varepsilon)$ .  $\square$

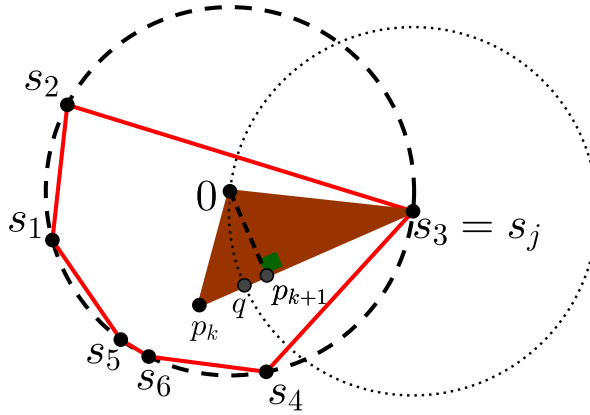


Figura 16 – Iteração do Algoritmo do Triângulo Esférico.

*Observação 4.6.* Analogamente ao que vimos no Teorema 3.8, quando  $0 \notin \text{conv}(\mathcal{S})$  a complexidade para encontrar uma testemunha só depende da geometria do problema, pois o mínimo de  $d(p', 0)$  para  $p' \in \text{conv}(\mathcal{S})$  será  $\Delta := \min\{d(x, 0) \mid x \in \text{conv}(\mathcal{S})\}$ . Note que  $R = 1$ , então, tomando  $\varepsilon = \Delta$ , sob as hipóteses do Teorema 4.5 em  $\mathcal{O}(M/\Delta)$  iterações, o TA-Esférico encontra uma testemunha.

*Observação 4.7.* Note que se  $M = 1/\varepsilon$ , qualquer  $p_k \in \text{conv}(\mathcal{S})$  satisfaz a  $\varepsilon/M$ -propriedade (enquanto não atingimos um dos critérios de parada), pois a  $\varepsilon/M$ -propriedade se reduz a  $\|p' - v\|^2 \geq 1 + \varepsilon^2$ , o que é automaticamente satisfeita ao escolher  $v$  como pivô estrito visto que

$$\|p' - v\|^2 = \|p'\|^2 + 1 - 2p'^T v \geq \|p'\|^2 + 1 > \varepsilon^2 + 1.$$

Observe que  $M = 1/\varepsilon$ , resulta na já esperada taxa de  $\mathcal{O}(1/\varepsilon^2)$ , logo a melhoria na complexidade é relativa, pois depende do quão grande é a constante  $M$ .

**Proposição 4.8.** Se  $\|p_k\| \geq \sqrt{\frac{\varepsilon}{M}}$  então para qualquer pivô estrito, o par  $(s_j, p_k)$  satisfaz a  $\varepsilon/M$ -propriedade.

*Demonstração.* Como  $s_j$  é um pivô estrito, temos que  $s_j^T p_k \leq 0$  e, por hipótese,  $\|p_k\| \geq \sqrt{\frac{\varepsilon}{M}}$ . Assim,

$$\|p_k\|^2 - 2s_j^T p_k \geq \|p_k\|^2 \geq \frac{\varepsilon}{M}.$$

Sabemos que  $\|s_j\| = 1$ , logo

$$\|p_k - s_j\|^2 = \|p_k\|^2 - 2s_j^T p_k + \|s_j\|^2 \geq 1 + \frac{\varepsilon}{M},$$

ou seja, o par  $(p_k, s_j)$  satisfaz a  $\varepsilon/M$ -propriedade.  $\square$

Esta última proposição mostra que a  $\varepsilon/M$ -propriedade é naturalmente satisfeita nas iterações iniciais do TA-esférico, para  $p_k$  suficientemente longe da origem. Se ao contrário das hipóteses da Proposição 4.8 vale  $\|p_k\| < \sqrt{\frac{\varepsilon}{M}}$ , nem sempre  $p_k$  admitirá um pivô  $s_j$  cumprindo (50). Em [42] é apresentada uma heurística para quando  $p_k$  não satisfizer a  $\varepsilon/M$ -propriedade. A ideia é encontrar um novo par  $(p_t, s)$  que garanta que  $\|p_t - s\|$  seja suficientemente grande com  $s$  pivô de  $p_t$ . No entanto, os autores de [42] não apresentam uma prova de corretude do procedimento interno, tampouco experimentos numéricos que evidenciem uma melhor complexidade de iteração.

Na próxima seção, vamos propor uma nova estratégia para encontrar um iterado  $p_k$  e pivô  $s$  que satisfaçam a  $\varepsilon/M$ -propriedade.

## 4.2 UMA NOVA HEURÍSTICA PARA O CHMP-ESFÉRICO

Considere  $p_k$  o iterado atual,  $s_j$  um pivô estrito para esse iterado e  $\theta_k := \angle p_k 0 s_j \in (\pi/2, \pi]$ . Observe que

$$\begin{aligned} \|p_k - s_j\|^2 &= \|p_k\|^2 + \|s_j\|^2 - 2p_k^T s_j \\ &= \|p_k\|^2 + \|s_j\|^2 - 2\|p_k\| \|s_j\| \cos \theta_k \\ &= 1 + \|p_k\| (\|p_k\| - 2 \cos \theta_k), \end{aligned} \tag{56}$$

visto que  $\|s_j\| = 1$ .

Assuma que o par  $(p_k, s_j)$  não satisfaz a  $\frac{\varepsilon}{M}$ -propriedade e que ainda não atingimos uma  $\varepsilon$ -solução. Assim, da Proposição 4.8, temos que  $\sqrt{\frac{\varepsilon}{M}} > \|p_k\| > \varepsilon$ . Disso, e de (56), queremos, se possível, garantir que  $\|p_k\| - 2 \cos \theta_k \geq \frac{1}{M}$ . Observe que quando  $M = 1$ , gostaríamos de ter  $\|p_k\| \geq 1 + 2 \cos \theta_k$ .

Nossa proposta é resolver um subproblema que “força”  $\cos \theta_k$  a ser o menor possível. Além disso, para esse procedimento, consideramos apenas os elementos de  $\mathcal{S}$  que já estão



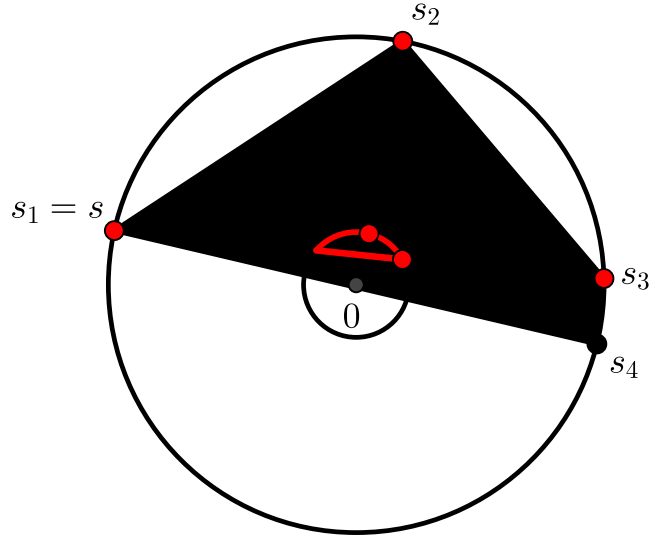


Figura 17 – Motivação para a nova heurística, em que  $p_k$  é combinação convexa de  $s_1, s_2, s_3$ ,  $s_1 = s$  é um pivô estrito para  $p_k$  e  $p_t = \tilde{S}x_t$  em que  $x_t$  é solução do problema (57).

ativos na combinação convexa que resulta em  $p_k$ . Para tanto, fixamos o pivô estrito  $s$  e obtemos um novo iterado  $p_t = \tilde{S}x_t$ , em que  $\tilde{S} \in \mathbb{R}^{m \times \ell}$  é uma matriz com os  $\ell$  elementos de  $\mathcal{S}$  que estão ativos na combinação convexa de  $p_k$  (o iterado atual) e  $x_t \in \Delta_\ell$  é tal que  $(\tilde{S}x_t)^T s$  seja o menor possível. Contudo, queremos preservar a distância de  $p_k$  para a origem, isto é, queremos que o novo iterado  $p_t$  seja tal que  $\|p_t\| \leq \|p_k\|$ , visto que nosso critério de parada é  $\|p_k\| \leq \varepsilon$ . Note que isso é equivalente a pedir que

$$\frac{1}{2}\|p_t\|^2 \leq \frac{1}{2}\|p_k\|^2.$$

Essa estratégia é ilustrada na Figura 17.

Assim, em cada iteração  $k$  do Algoritmo 7, na qual  $p_k$ , com o respectivo pivô estrito  $s_j$ , não satisfazem a  $\frac{\varepsilon}{M}$ -propriedade, iremos resolver o subproblema a seguir

$$\begin{aligned} \min_x \quad & (\tilde{S}x)^T s_j, \\ \text{s. a} \quad & e^T x = 1, \\ & x \geq 0, \\ & \frac{1}{2}\|\tilde{S}x\|^2 \leq \frac{1}{2}\|p_k\|^2, \end{aligned} \tag{57}$$

em que  $\tilde{S} \in \mathbb{R}^{m \times \ell}$  é uma matriz com os  $\ell$  elementos de  $\mathcal{S}$  que estão ativos na combinação convexa de  $p_k$  (o iterado atual), isto é,  $p_k = \tilde{S}\tilde{\alpha}$  com  $e^T \tilde{\alpha} = 1$  e  $\tilde{\alpha} \in \mathbb{R}_+^\ell$ . Note que (57) é viável, pois  $\tilde{\alpha}$  satisfaz as suas restrições.

O problema (57) é um problema de otimização não-linear, em que somente uma das restrições é de fato não-linear. Martein e Schaible [49] apresentaram uma versão mais

geral desse problema, e um dos resultados auxiliares presente no artigo é interessante para uma análise que faremos posteriormente.

Para ficarmos condizentes com a notação de [49], considere o problema de programação linear

$$\begin{aligned} \min_x \quad & (\tilde{S}x)^T s_j, \\ \text{s. a} \quad & e^T x = 1, \\ & x \geq 0, \end{aligned} \tag{58}$$

e defina  $S_L$  como o conjunto de soluções ótimas para (58). Além disso, considere

$$K := \{x \in \mathbb{R}^\ell \mid \|\tilde{S}x\| \leq \|p_k\|\}.$$

O resultado a seguir nos mostra que a solução do problema (57) satisfaz a restrição quadrática na igualdade.

**Teorema 4.9** ([49, Theorem 1.1(iii)]). *Se  $S_L \cap K = \emptyset$ , então qualquer solução  $x^*$  do problema (57) é tal que  $\|\tilde{S}x^*\| = \|p_k\|$ .*

*Demonstração.* Seja  $x_L \in S_L$  solução de (58), isto é,

$$(\tilde{S}x_L)^T s_j \leq (\tilde{S}x)^T s_j \quad \text{para todo } x \text{ tal que } e^T x = 1, x \geq 0. \tag{59}$$

Considere  $x^*$  solução de (57). Observe que a igualdade em (59) não pode ser atingida para  $x = x^*$  pois implicaria em  $x^* \in S_L$ , contradizendo a hipótese  $S_L \cap K = \emptyset$ . Portanto

$$(\tilde{S}x_L)^T s_j < (\tilde{S}x^*)^T s_j. \tag{60}$$

Seja  $d := x_L - x^* \neq 0$ . De (60) deduzimos que  $d$  é uma direção de descida a partir de  $x^*$ . Agora suponha que  $\|Sx^*\| < \|p_k\|$ , vamos provar que  $d$  é uma direção factível a partir de  $x^*$ . É claro que para  $t > 0$ , suficientemente pequeno,  $\hat{x} = x^* + t(x_L - x^*)$  satisfaz  $\hat{x}^T e = 1$ ,  $\hat{x} \geq 0$ , uma vez que  $\hat{x}$  é combinação convexa de pontos no simplex unitário. Note que

$$\begin{aligned} \|\tilde{S}\hat{x}\|^2 &= \|\tilde{S}(x^* + t(x_L - x^*))\|^2 \\ &= \|\tilde{S}x^*\|^2 + 2t(\tilde{S}x^*)^T \tilde{S}(x_L - x^*) + t^2 \|\tilde{S}(x_L - x^*)\|^2. \end{aligned}$$

Ou seja, temos uma quadrática em relação a  $t$  com o termo independente  $\|\tilde{S}x^*\|^2 < \|p_k\|^2$ . Além disso,  $\|\tilde{S}(x_L - x^*)\|^2 > 0$ , ou seja a concavidade da parábola é positiva. Logo, para  $t$  suficientemente pequeno,  $\|\tilde{S}\hat{x}\|^2 \leq \|p_k\|^2$ . Disso, e de (60), temos que  $d$  é uma direção factível e de descida a partir de  $x^*$  o que contradiz o fato de  $x^*$  ser minimizador de (57). Sendo assim,  $x^*$  é tal que  $\frac{1}{2} \|\tilde{S}x^*\|^2 = \frac{1}{2} \|p_k\|^2$ .

□

*Observação 4.10.* De fato, caso o iterado  $p_k$ , com o respectivo pivô  $s$ , não satisfaça a  $\frac{\varepsilon}{M}$ -propriedade temos que  $S_L \cap K = \emptyset$ , uma vez que em tal cenário  $\varepsilon < \|p_k\| < \sqrt{\frac{\varepsilon}{M}}$  e analisando o conjunto solução de (PL) temos que  $e_j \in S_L$ , em que  $j \in \arg \min \{s_i^T s_j \mid 1 \leq i \leq \ell\}$ . Assim,  $\|\tilde{S}e_j\| = 1 > \|p_k\|$ , ou seja,  $e_j \notin K$ .

Os próximos resultados nos motivam a encontrar uma solução  $x^*$  de (57), pois assim obtemos um novo iterado  $p_t := \tilde{S}x^*$  tal que  $\|p_t - s\|^2 \geq \|p_k - s\|^2$ .

**Proposição 4.11.** *Seja  $p_t := \tilde{S}x^*$ , em que  $x^*$  é solução de (57). Então  $\|p_t - s_j\|^2 \geq \|p_k - s_j\|^2$ .*

*Demonstração.* Do Teorema 4.9, temos que qualquer solução  $x^*$  de (57) é tal que  $\|\tilde{S}x^*\| = \|p_t\| = \|p_k\|$ . Além disso, sabemos que  $p_t^T s_j \leq p_k^T s_j$  pois  $p_k = \tilde{S}\alpha$  e  $\alpha$  é viável para (57). Dessa inequação temos

$$(p_k - p_t)^T s_j \geq 0. \quad (61)$$

Então, utilizando (61), segue que

$$\begin{aligned} \|p_t - s_j\|^2 - \|p_k - s_j\|^2 &= \|s_j\|^2 - 2s_j^T p_t + \|p_t\|^2 - \|s_j\|^2 + 2s_j^T p_k - \|p_k\|^2 \\ &= 2(p_k - p_t)^T s_j \\ &\geq 0. \end{aligned}$$

□

**Colorário 4.12.** *Seja  $x^*$  solução do subproblema (57). Se existe  $\hat{p} \in \mathbb{R}^m$  tal que  $\|\hat{p} - s_j\|^2 \geq 1 + \varepsilon/M$ , com  $\hat{p} := \tilde{S}x$ ,  $x \geq 0$ ,  $e^T x = 1$  e  $\|\hat{p}\| \leq \|p_k\|$ , então  $p_t := \tilde{S}x^*$  satisfaz a  $\varepsilon/M$ -propriedade.*

*Demonstração.* Do Teorema 4.9 temos que  $\|p_t\| = \|p_k\|$ . Além disso, como  $\hat{p}$  é um ponto viável para (57) temos que:

$$p_t^T s_j \leq \hat{p}^T s_j.$$

Logo:

$$\|p_t - s_j\|^2 - \|\hat{p} - s_j\|^2 = \|p_t\|^2 - \|\hat{p}\|^2 + 2(\hat{p} - p_t)^T s_j \geq 0.$$

Com isso,  $\|p_t - s_j\|^2 \geq \|\hat{p} - s_j\|^2$  e, portanto,  $p_t$  satisfaz a  $\frac{\varepsilon}{M}$ -propriedade. □

Portanto, o procedimento a ser integrado ao TA-Esférico, como apresentado no Algoritmo 8, fornece um novo iterado  $p_t$  que satisfaz a  $\varepsilon/M$ -propriedade, ou, que pelo menos cumpre  $\|p_t - s_j\|^2 \geq \|p_k - s_j\|^2$ . Dessa maneira, chamamos tal procedimento de heurística, na medida em que, não temos garantias teóricas de que o novo par  $(p_t, s_j)$  irá satisfazer a  $\varepsilon/M$ -propriedade.

Tanto quanto sabemos, não há uma solução analítica para o problema (57). Na próxima seção discutiremos um método para resolução aproximada de tal subproblema.

**Algoritmo 8:** ALGORITMO DO TRIÂNGULO ESFÉRICO COM HEURÍSTICA (TA-ESFÉRICOH)

---

**Data:**  $S, \varepsilon \in (0, 1)$

- 1 Escolha  $p_0 = s_1$
- 2 **para**  $k = 0, 1, 2, \dots$  **faça**
- 3     **se**  $\|p_k\| < \varepsilon$  **então**
- 4         | pare
- 5     **fim**
- 6     **se**  $s_j^T p_k > 0, \forall j = 1, \dots, n$  **então**
- 7         | pare:  $p_k$  é uma  $p$ -testemunha.
- 8     **senão**
- 9         | Escolha  $s_j \neq p_k$ , tal que  $s_j^T p_k \leq 0$ .
- 10         | **se**  $\|p_k - s_j\|^2 < 1 + \varepsilon/M$  **então**
- 11             | Faça  $p_k \leftarrow \tilde{S}x^*$ , em que  $x^*$  é solução de (57)
- 12         | **fim**
- 13     **fim**
- 14      $p_{k+1} \leftarrow p_k + \frac{p_k^T (s_j - p_k)}{\|p_k - s_j\|^2} (p_k - s_j)$ .
- 15 **fim**

---

## 4.3 RESOLVENDO O SUBPROBLEMA

Começamos esta seção apresentando as condições KKT do problema (57). Considere  $x \in \mathbb{R}^\ell$  um ponto que satisfaz as condições KKT, com multiplicadores  $\lambda \in \mathbb{R}$ ,  $\mu \in \mathbb{R}^\ell$  e  $\eta \in \mathbb{R}$ , isto é, que sejam válidas

$$\begin{aligned}
& \tilde{S}^T s_j + \lambda e + \eta \tilde{S}^T \tilde{S}x - \mu = 0, \\
& e^T x = 1, \quad x \geq 0, \quad \mu \geq 0, \quad \eta \geq 0, \\
& \frac{1}{2} \|\tilde{S}x\|^2 \leq \frac{1}{2} \|p_k\|^2 \\
& \eta \left( \frac{1}{2} \|\tilde{S}x\|^2 - \frac{1}{2} \|p_k\|^2 \right) = 0, \\
& \mu^T x = 0.
\end{aligned} \tag{62}$$

Agora exploraremos o dual do problema (57), levando em conta apenas a restrição quadrática, deixando as restrições que definem o simplex unitário como restrições de domínio da função objetivo. Para isso, definimos a função Lagrangiana associada, com  $\eta \geq 0$  e  $x \in \Delta_\ell$  por

$$L(x, \eta) := (\tilde{S}x)^T s_j + \frac{\eta}{2} \left( \|\tilde{S}x\|^2 - \|p_k\|^2 \right).$$

Logo, para  $\eta > 0$ ,

$$\begin{aligned} L(x, \eta) &= \eta \left( -(\tilde{S}x)^T(-s_j/\eta) + \frac{1}{2} \left( \|\tilde{S}x\|^2 - \|p_k\|^2 \right) + \frac{\|s_j\|^2}{2\eta^2} - \frac{\|s_j\|^2}{2\eta^2} \right) \\ &= \eta \left( \frac{1}{2} \|\tilde{S}x + \frac{1}{\eta} s_j\|^2 - \frac{\|p_k\|^2}{2} - \frac{1}{2\eta^2} \right). \end{aligned}$$

Considere a função dual definida pelo subproblema de otimização:

$$\begin{aligned} \phi(\eta) &:= \min_x L(x, \eta) \\ \text{s. a } &e^T x = 1, \\ &x \geq 0 \end{aligned} \tag{63}$$

Para cada  $\eta > 0$  fixo, o problema que define  $\phi(\eta)$  em (63) equivale a:

$$\begin{aligned} \min_x & \frac{1}{2} \|\tilde{S}x + \frac{1}{\eta} s_j\|^2 \\ \text{s. a } & e^T x = 1, \\ & x \geq 0 \end{aligned} \tag{64}$$

Observe que (64) está associado ao seguinte CHMP: determinar se  $-s_j/\eta$  pertence ao envoltório convexo das colunas de  $\tilde{S}$ . Além disso, as condições KKT para (64) são:

$$\begin{aligned} \frac{1}{\eta} \tilde{S}^T s_j + \hat{\lambda} e + \tilde{S}^T \tilde{S}x - \hat{\mu} &= 0, \\ e^T x = 1, \quad x \geq 0, \quad \hat{\mu} &\geq 0, \\ \hat{\mu}_i x_i &= 0, \quad \forall i, \end{aligned}$$

para  $\hat{\lambda} \in \mathbb{R}$ ,  $\hat{\mu} \in \mathbb{R}^\ell$ . Dessa forma, fica claro que se  $x^*$  é uma solução de (64) com  $\|\tilde{S}x^*\| = \|p_k\|$ , as condições KKT de (64) equivalem as condições KKT de (57) com  $\lambda = \eta \hat{\lambda}$  e  $\mu = \eta \hat{\mu}$ .

Nossa estratégia para resolver o subproblema (57) é resolver (64) variando  $\eta$  até que  $\|\tilde{S}x^*\| = \|p_k\|$ . Usaremos para tanto o método de Uzawa [1, 19, 44] que, grosso modo, a fim de encontrar um ponto sela do Lagrangiano (64), aplica um método de máxima subida à função dual  $\phi(\eta)$ . Para  $\eta_t > 0$  fixo, resolve-se (64), cuja solução determina  $x_t$ . A seguir, o multiplicador é atualizado da forma  $\eta_{t+1} := \max\{\eta_t + \frac{\delta}{2}(\|\tilde{S}x_t\|^2 - \|p_k\|^2), 0\}$ . Este processo é resumido no Algoritmo 9.

*Observação 4.13.* Seja  $w$  tal que  $(w - s_j) \in \text{Im}(\tilde{S})$ . Perceba que escolhendo  $\lambda = -(\tilde{S}e_j)^T w$  em que  $j = \arg \min (\tilde{S}e_i)^T w$ ,  $\mu_i = (\tilde{S}e_i)^T w_i - (\tilde{S}e_j)^T w$  para  $i \neq j$  e  $\mu_j = 0$ , temos que:

$$(-\lambda e + \mu) = \tilde{S}^T w.$$

De (62) temos que  $\eta \tilde{S}^T \tilde{S}x = -\lambda e + \mu - \tilde{S}^T s_j$  e assim

$$\eta \tilde{S}^T \tilde{S}x = \tilde{S}^T (w - s_j).$$

**Algoritmo 9:** MÉTODO DE UZAWA PARA O SUBPROBLEMA (57)

---

**Dados:**  $\eta_0 \in \mathbb{R}^n, \delta > 0, r = \|p_k\|, \varepsilon > 0.$

- 1 **para**  $t = 0, 1, 2, \dots$  **faça**
- 2     Encontre  $x_t$  solução de (64);
- 3     Defina  $q_t := \tilde{S}x_t$ ;
- 4     **se**  $|\|q_t\|^2 - r^2| \leq \varepsilon$  **então**
- 5         pare
- 6     **fim**
- 7      $\eta_{t+1} := \max \left\{ \eta_t + \frac{\delta}{2} (\|q_t\|^2 - r^2), 0 \right\}$
- 8 **fim**

---

Como  $(w - s_j) \in \text{Im}(\tilde{S})$ , existe  $x$  tal que

$$\tilde{S}x = \frac{1}{\eta}(w - s_j).$$

Note que se  $\eta = \frac{\|w - s_j\|}{\|p_k\|}$  temos que  $\|\tilde{S}x\| = \|p_k\|$ . Infelizmente não temos garantia de que tal  $x$  satisfaz as outras condições de (62), mas esta análise nos motivou a escolher  $\eta_0$  como  $\mathcal{O}\left(\frac{1}{\|p_k\|}\right)$  nos experimentos numéricos.

#### 4.4 EXPERIMENTOS NUMÉRICOS PARA O TA-ESFÉRICO

Nessa seção apresentamos dois conjuntos de experimentos computacionais para resolução do CHMP Esférico.

Na Seção 4.4.1, temos como objetivo principal investigar os resultados de complexidade de iteração do TA-Esférico em comparação ao TA-EsféricoH (Algoritmo 8). Posteriormente, na Seção 4.4.2, consideramos problemas gerados aleatoriamente para investigar a eficiência da heurística proposta em termos de tempo computacional.

Novamente todos os testes foram executados em computador pessoal com processador Intel Core i7-8565U 1.80 GHz com 8 GB de RAM rodando Windows 10. Os algoritmos foram implementados no Matlab R2019a. Os códigos dos experimentos dessa seção estão disponíveis em [https://github.com/rafaelafilipozzi/Spherical\\_TA](https://github.com/rafaelafilipozzi/Spherical_TA).

##### 4.4.1 Investigação empírica da complexidade de iteração

Nessa seção queremos investigar os resultados de complexidade de iteração vistos na Seção 4.1. Para isso geramos  $\mathcal{A} := \{v_1, \dots, v_n\}$  como na Seção 3.6.1 e  $p$  como no cenário (b) (Seção 3.6.1.2), pois foi o cenário mais desafiador para o Algoritmo do Triângulo. Dessa forma o problema de inclusão no envoltório convexo esférico que iremos analisar é para  $\mathcal{S} = \{s_1, \dots, s_n\} \subset \mathbb{R}^m$ , com  $s_i := (v_i - p) / \|v_i - p\|$ .

Lembramos que, do Teorema 3.8, a complexidade de iteração do TA-Esférico, que é um caso específico do TA, é

$$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right).$$

Contudo, como vimos no Teorema 4.5, se a cada iteração o par  $(p_k, s_j)$  (quando  $p_k$  não é uma testemunha), satisfaz a  $\varepsilon/M$ -propriedade, temos que a complexidade de iteração do TA-Esférico para encontrar uma  $p_\varepsilon$ -solução é

$$\mathcal{O}\left(\frac{M}{\varepsilon}\right).$$

Da Observação 4.7 temos uma melhoria na complexidade de iteração, desde que  $1 \leq M < \frac{1}{\varepsilon}$ . Nos testes computacional consideramos  $M = 1$ .

Salientamos que o passo 11 do TA-EsféricoH (Algoritmo 8) foi resolvido utilizando o Método de Uzawa para o subproblema (57) (Algoritmo 9). Por sua vez para a resolução do problema (64) utilizamos o solver `quadprog` do Matlab. Além disso, optamos por definir o número máximo de iterações como  $\frac{1}{\varepsilon^2}$  com base na complexidade de iteração para o TA-Esférico. Foram gerados 10 exemplares aleatórios (como descrito no início da seção) fixando a dimensão como  $m = 50$  e o número de pontos como  $n = 500$  e variando a tolerância  $\varepsilon$  de  $10^{-5}$  a  $10^{-3}$ .

A Figura 18 mostra o desempenho do TA-Esférico e TA-EsféricoH em termos de número médio de iterações (em escala logarítmica) para encontrar um ponto tal que  $\|p_\varepsilon\| \leq \varepsilon$ , variando  $\varepsilon$ . Para referência acrescentamos as curvas de  $(\log_{10} \text{ de}) 1/\varepsilon$  e  $1/\varepsilon^2$ . Observamos que a heurística proposta promoveu uma melhoria na complexidade, de  $\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$  para  $\mathcal{O}\left(\frac{M}{\varepsilon}\right)$ , mesmo não tendo garantia teórica de gerar iterados “corrigidos” satisfazendo a  $\varepsilon/M$ -propriedade.

Observamos que em exemplares nos quais a heurística foi acionada, ou seja,  $\varepsilon/M$ -propriedade não foi satisfeita em alguma iteração, em no máximo 3 iterações após o acionamento, o TA-EsféricoH encontrou uma  $p_\varepsilon$ -solução (e nestas iterações a heurística também foi acionada).

Vale ressaltar que o Algoritmo do TA-Esférico e do TA-EsféricoH, coincidem nas iterações iniciais, enquanto a  $\varepsilon/M$ -propriedade é satisfeita. Por exemplo, para um exemplar com  $\varepsilon = 0.001$ , o TA-EsféricoH acionou a heurística 3 vezes a partir da iteração 750 e parou após a iteração 752, enquanto o TA-Esférico parou somente na iteração 249997.

Além disso, todas as vezes que o par  $(p_k, s_j)$  não satisfaz a  $\varepsilon/M$ -propriedade, a solução do subproblema resolvido pelo Algoritmo 9 com  $\eta_0 = \frac{1}{\|p_k\|}$ , em apenas uma iteração, resultou no par  $(p_t, s_j)$  satisfazendo a  $\varepsilon/M$ -propriedade, em que  $p_t = \tilde{S}x_t$ .

#### 4.4.2 Exemplares artificiais para o CHMP Esférico e tempo de execução

Nessa seção optamos por gerar os exemplares como em [42] para comparar tempo computacional do TA-Esférico e TA-EsféricoH. Foram gerados 10 exemplares aleatórios

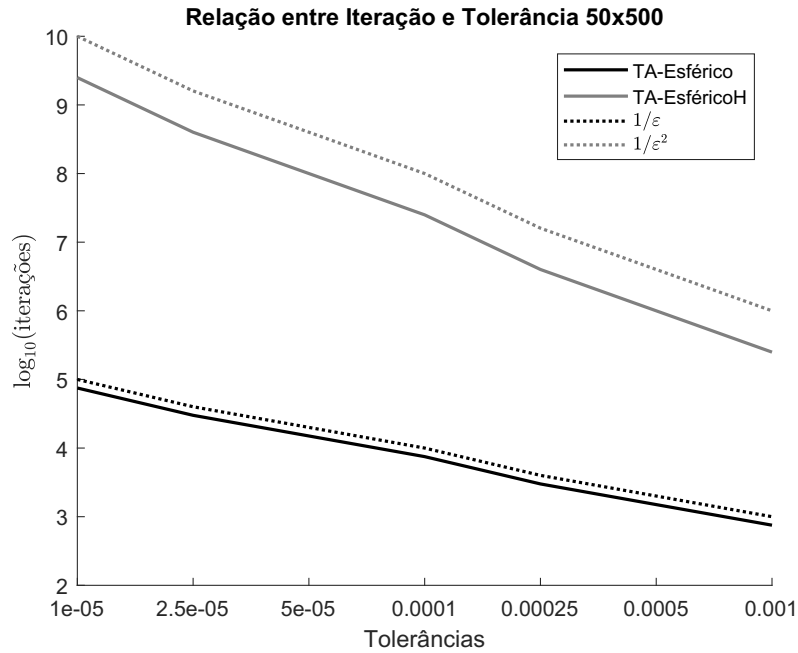


Figura 18 – Relação entre o número de iterações conforme aumentamos a tolerância para a dimensão  $m = 50$  e  $n = 500$ .

para cada dimensão variando de  $m = 50, n = 500$  a  $m = 300, n = 3000$ , como especificado na Tabela 5.

A Figura 19 apresenta o tempo computacional para  $\varepsilon = 10^{-7}$  e na Tabela 5 temos o número de iterações (em média) para finalização do TA-Esférico e TA-EsféricoH. No caso em que  $0 \in \text{conv}(\mathcal{S})$  e as dimensões são menores o TA-EsféricoH possui um desempenho um pouco pior que o TA-Esférico em relação ao tempo computacional, porém necessita de menos iterações para encontrar uma  $p_\varepsilon$ -solução. Isto ocorre pelo custo de resolver (64), pois até o momento não encontramos uma maneira analítica de resolver tal problema e optamos por utilizar o solver `quadprog` do Matlab.

Já no caso de encontrar uma testemunha de que  $0 \notin \text{conv}(\mathcal{S})$ , ambos algoritmos detectaram uma testemunha em poucas iterações e com tempo computacional bem baixo. Além disso, nesse caso o TA-EsféricoH não acionou nenhuma vez a heurística proposta.

Salientamos que novamente nos casos em que TA-EsféricoH acionou a heurística, a  $\varepsilon/M$ -propriedade foi satisfeita para o novo par  $(p_t, s_j)$  com  $p_t = \tilde{S}x_t$  sendo  $x_t$  solução de (64). Além disso, nesses casos a heurística foi acionada apenas uma vez, na iteração anterior a encontrar uma  $\varepsilon$ -solução.

#### 4.5 CONCLUSÕES DO CAPÍTULO

Neste capítulo apresentamos um procedimento heurístico integrado ao Algoritmo do Triângulo Esférico a fim de promover a  $\varepsilon/M$ -propriedade a cada iteração. Embora não tenhamos comprovação teórica de que o novo ponto  $p_t$ , juntamente com o pivô  $s_j$ , irá satisfazer a  $\varepsilon/M$ -propriedade, nos experimentos numéricos percebemos que isso ocorreu



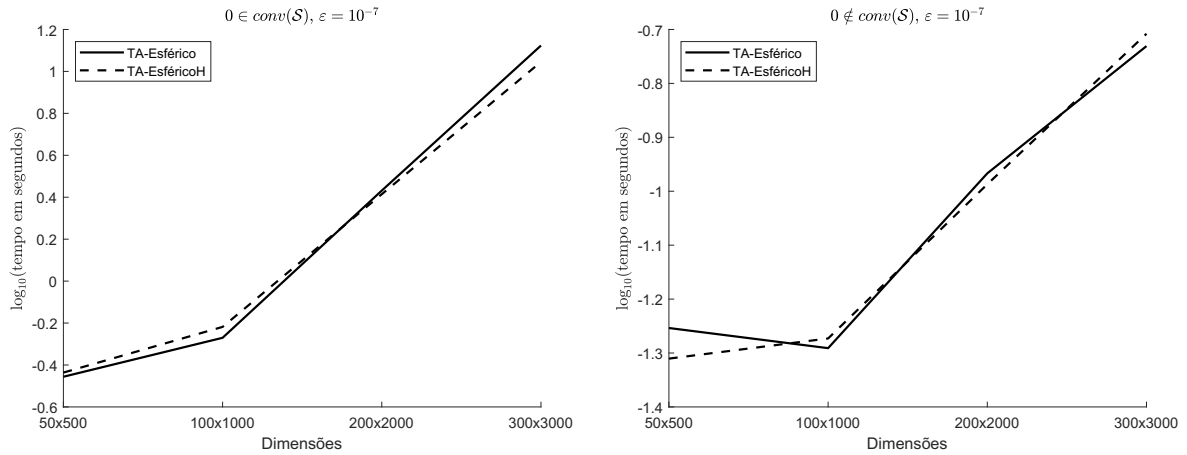


Figura 19 – Tempos de execução (em segundos) para o CHMP-Esférico.

Tabela 5 – Média de iterações dos experimentos com CHMP-Esférico

$m$	$n$	$0 \in \text{conv}(\mathcal{S})$		$0 \notin \text{conv}(\mathcal{S})$	
		TA-Esférico	TA-EsféricoH	TA-Esférico	TA-EsféricoH
50	500	1512.3	1272.0	128.8	128.8
100	1000	2949.7	2403.8	60.6	60.6
200	2000	5561.2	4336.7	15.8	15.8
300	3000	8560.3	6400.0	7.6	7.6

em todos os exemplares (que precisaram realizar esse procedimento). Portanto temos a seguinte conjectura.

*Conjectura 4.14.* Considere que na  $k$ -ésima iteração do Algoritmo do Triângulo Esférico o iterado  $p_k$  é tal que  $\varepsilon \leq \|p_k\| \leq \sqrt{\varepsilon/M}$ . Seja  $s_j$  um pivô para  $p_k$  e assumamos que o par  $(p_k, s_j)$  não satisfaz a  $\varepsilon/M$ -propriedade. Então o par  $(p_t, s_j)$ , em que  $p_t := \tilde{S}x_t$  com  $x_t$  solução de (57), satisfaz a  $\varepsilon/M$ -propriedade.

Percebemos que para encontrar uma  $p_\varepsilon$ -solução o número de iterações do TA-EsféricoH diminui consideravelmente em relação ao TA-Esférico, principalmente em casos de maior dificuldade como na Seção 4.4.1. Como trabalho futuro pretendemos otimizar a resolução do problema (64), encontrando, se possível, uma forma analítica, pois dessa maneira o TA-EsféricoH se tornará mais competitivo em relação ao tempo computacional.

## 5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Na primeira parte deste trabalho propusemos a aplicação de métodos do tipo Frank-Wolfe e Gradiente Projetado para o problema de inclusão no envoltório convexo. Para isso, desenvolvemos critérios de parada adequados, que permitiram a resolução eficiente do problema de decisão associado. Experimentos numéricos considerando vários cenários diferentes do CHMP indicaram um melhor desempenho do ASFW quando comparado com o algoritmo do Triângulo (e variantes), que é específico para o problema, especialmente em exemplares mais difíceis, com geometria desfavorável, como os casos das Seções 3.6.1.2 e 3.6.1.4.

Dentre as aplicações, consideramos problemas de factibilidade em programação linear que podem ser formulados como um CHMP. Os resultados indicam que os métodos de primeira ordem podem ser superiores ao algoritmo dual simplex clássico e o Algoritmo do Triângulo Ganancioso aparenta ser uma alternativa eficiente para detectar infactibilidades. Apesar do ASFW possuir taxa de convergência linear, a geometria do problema nesse caso influencia negativamente em sua convergência resultando em um desempenho inferior nos casos de factibilidade. Observamos que o desempenho dos algoritmos do tipo Frank-Wolfe em instâncias factíveis de problemas de factibilidade em PL foi diferente do que observamos nas Seções 3.6.1.1 e 3.6.1.2. Conforme discutido na Seção 3.6.2, isso tem a ver com o condicionamento do problema (que depende da geometria do envoltório convexo), o que torna as coisas mais difíceis mesmo para o ASFW que possui convergência linear. Em trabalhos futuros planejamos investigar isso precisamente, bem como possíveis estratégias de aceleração para métodos do tipo FW aplicados ao CHMP.

Também apresentamos a aplicação do CHMP para o problema de classificação de imagens. Vimos que utilizar a distância de um ponto teste a uma testemunha ao invés da distância do ponto ao envoltório convexo não altera consideravelmente a precisão da classificação. Mais ainda, vimos que os métodos de primeira ordem específicos para o CHMP resolveram esses problemas cerca de vinte vezes mais rápido do que computando a projeção exata.

Para investigação futura pretendemos aplicar o CHMP e os métodos desenvolvidos nesta tese para a classificação de imagens em outros conjuntos de dados. Além disso, buscaremos considerar outras variantes do método de Frank-Wolfe, como *pairwise* [45] e *fully corrective* [35], para o CHMP, com o intuito de ter um estudo numérico mais aprofundado.

Na segunda parte deste trabalho apresentamos o problema esférico de inclusão no envoltório convexo e um algoritmo do Triângulo adequado a este problema, o TA-Esférico [43]. Propomos uma heurística que visa melhorar a complexidade de iteração deste algoritmo de  $\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$  para  $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ . Experimentos preliminares indicam que tal heurística cumpriu bem seu papel. Contudo, ao analisarmos a questão do tempo computacional

não notamos uma diferença significativa entre o algoritmo clássico e o algoritmo com a nova heurística. A razão é o custo computacional na resolução do subproblema quadrático demandado pela nova heurística. Em um estudo futuro pretendemos usar métodos ADMM (Alternating Direction Method of Multipliers) [25, 26] ou métodos PDHG (Primal-Dual Hybrid Gradient) [12] para a resolução eficiente de tal subproblema.

Por fim, em todos os experimentos numéricos realizados observamos que a heurística foi capaz de gerar iterados satisfazendo a  $\varepsilon/M$ -propriedade, que garante a complexidade  $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ , embora não tenhamos uma demonstração formal para este fato. Deixamos então esta questão como uma conjectura.

Vale ressaltar novamente que boa parte do Capítulo 3 foi publicada no artigo [22]. Mais ainda, a autora dessa tese apresentou, parte de tal capítulo na “International Conference on Continuous Optimization (ICCOPT)” em Bethlehem, Pennsylvania, 2022, assim como parte do Capítulo 4 foi apresentado no Congresso Nacional de Matemática Aplicada e Computacional (CNMAC), virtualmente em 2022.

## REFERÊNCIAS

- [1] G. Allaire. *Numerical analysis and optimization: an introduction to mathematical modelling and numerical simulation*. OUP Oxford, 2007 (ver p. 68).
- [2] P. Awasthi, B. Kalantari e Y. Zhang. “Robust vertex enumeration for convex hulls in high dimensions”. *In: Annals of Operations Research* 295.1 (2020), pp. 37–73 (ver pp. 10, 12).
- [3] C. B. Barber, D. P. Dobkin e H. Huhdanpaa. “The quickhull algorithm for convex hulls”. *In: ACM Transactions on Mathematical Software (TOMS)* 22.4 (1996), pp. 469–483 (ver p. 10).
- [4] J. Barzilai e J. M. Borwein. “Two-point step size gradient methods”. *In: IMA Journal of Numerical Analysis* 8.1 (1988), pp. 141–148 (ver p. 26).
- [5] M. S. Bazaraa, J. J. Jarvis e H. D. Sherali. *Linear Programming and Network Flows*. Fourth. New York: Wiley, dez. de 2009 (ver p. 14).
- [6] A. Beck e S. Shtern. “Linearly Convergent Away-Step Conditional Gradient for Non-Strongly Convex Functions”. *In: Mathematical Programming* 164.1 (jul. de 2017), pp. 1–27 (ver p. 24).
- [7] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific Belmont, 1999 (ver pp. 14–19, 24).
- [8] E. G. Birgin, J. M. Martínez e M. Raydan. “Spectral projected gradient methods”. en. *In: Encyclopedia of Optimization*. Ed. por C. A. Floudas e P. M. Pardalos. Boston, MA: Springer US, 2009, pp. 3652–3659 (ver pp. 12, 25, 26).
- [9] E. G. Birgin, J. M. Martínez e M. Raydan. “Nonmonotone Spectral Projected Gradient Methods on Convex Sets”. *In: SIAM Journal on Optimization* 10.4 (jan. de 2000), pp. 1196–1211 (ver pp. 12, 26).
- [10] E. G. Birgin, J. M. Martínez e M. Raydan. “Spectral Projected Gradient Methods: Review and Perspectives”. *In: Journal of Statistical Software* 60.3 (2014) (ver pp. 25, 26).

- 
- [11] M. D. Canon e C. D. Cullum. “A Tight Upper Bound on the Rate of Convergence of Frank-Wolfe Algorithm”. In: *SIAM Journal on Control* 6.4 (nov. de 1968), pp. 509–516 (ver p. 21).
- [12] A. Chambolle e T. Pock. “A first-order primal-dual algorithm for convex problems with applications to imaging”. In: *Journal of mathematical imaging and vision* 40 (2011), pp. 120–145 (ver p. 74).
- [13] L.-L. Chen e T. C. Woo. “Computational Geometry on the Sphere With Application to Automated Machining”. In: *Journal of Mechanical Design* 114.2 (jun. de 1992), pp. 288–295 (ver p. 10).
- [14] K. L. Clarkson. “Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm”. In: *ACM Transactions on Algorithms* 6.4 (ago. de 2010), pp. 1–30 (ver p. 20).
- [15] C. Combettes e S. Pokutta. “Boosting Frank-Wolfe by chasing gradients”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 2111–2121 (ver pp. 11, 19).
- [16] L. Condat. “Fast Projection onto the Simplex and the  $\ell_1$  ball”. In: *Mathematical Programming* 158.1 (jul. de 2016), pp. 575–585 (ver pp. 11, 20, 43).
- [17] Y.-H. Dai. “On the nonmonotone line search”. In: *Journal of Optimization theory and Applications* 112.2 (2002), pp. 315–330 (ver p. 26).
- [18] G. B. Dantzig. *An  $\epsilon$ -precise feasible solution to a linear program with a convexity constraint in  $\frac{1}{\epsilon^2}$  iterations independent of problem size*. Tech. rep. Stanford University, 1992 (ver pp. 11, 41).
- [19] A. Duguid. “Studies in linear and non-linear programming, by KJ Arrow, L. Hurwicz and H. Uzawa. Stanford University Press, 1958. 229 pages.” In: *Canadian Mathematical Bulletin* 3.3 (1960), pp. 196–198 (ver p. 68).
- [20] M. Epeleman e R. M. Freund. “Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system”. In: *Mathematical Programming* 88.3 (2000), pp. 451–485 (ver p. 41).

- [21] R. Filippozzi. “Um Algoritmo Geométrico Para o Problema de Inclusão No Envoltório Convexo”. Diss. de mestr. Florianópolis: Matemática Pura e Aplicada/Universidade Federal de Santa Catarina, 2019 (ver pp. 29, 31, 34).
- [22] R. Filippozzi, D. S. Gonçalves e L.-R. Santos. “First-Order Methods for the Convex Hull Membership Problem”. In: *European Journal of Operational Research* 306.1 (2023), pp. 17–33 (ver pp. 12, 28, 74).
- [23] M. Frank e P. Wolfe. “An Algorithm for Quadratic Programming”. In: *Naval Research Logistics Quarterly* 3.1-2 (mar. de 1956), pp. 95–110 (ver pp. 11, 19).
- [24] A. Friedlander. *Elementos de Programação Não-Linear*. Campinas: IMECC/Unicamp, 1994 (ver p. 17).
- [25] D. Gabay e B. Mercier. “A dual algorithm for the solution of nonlinear variational problems via finite element approximation”. In: *Computers & mathematics with applications* 2.1 (1976), pp. 17–40 (ver p. 74).
- [26] R. Glowinski e A. Marroco. “Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires”. In: *Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique* 9.R2 (1975), pp. 41–76 (ver p. 74).
- [27] J. Gonçalves, R. Storer e J. Gondzio. “A family of linear programming algorithms based on an algorithm by von Neumann”. In: *Optimization Methods and Software* 24.3 (2009), pp. 461–478 (ver p. 41).
- [28] R. L. Graham. “An efficient algorithm for determining the convex hull of a finite planar set”. In: *Information Processing Letters* 1 (1972), pp. 132–133 (ver p. 10).
- [29] G. N. Grapiglia e E. W. Sachs. “On the Worst-Case Evaluation Complexity of Non-Monotone Line Search Algorithms”. In: *Computational Optimization and Applications* 68.3 (dez. de 2017), pp. 555–577 (ver pp. 11, 26, 27).
- [30] L. Grippo, F. Lampariello e S. Lucidi. “A Nonmonotone Line Search Technique for Newton’s Method”. en. In: *SIAM Journal on Numerical Analysis* 23.4 (ago. de 1986), pp. 707–716 (ver pp. 11, 26).

- [31] J. GuéLat e P. Marcotte. “Some comments on Wolfe’s ‘away step’”. In: *Mathematical Programming* 35.1 (mai. de 1986), pp. 110–119 (ver pp. [22](#), [40](#)).
- [32] R. Harman e V. Lacko. “On Decompositional Algorithms for Uniform Sampling from  $n$ -Spheres and  $n$ -Balls”. In: *Journal of Multivariate Analysis* 101.10 (nov. de 2010), pp. 2297–2304 (ver p. [46](#)).
- [33] A. Izmailov e M. Solodov. *Otimização, Volume I - Condições de Otimalidade. Elementos de Análise Convexa e de Dualidade*. 3<sup>a</sup> ed. Vol. 1. 2 vol. Rio de Janeiro: IMPA, 2007 (ver p. [14](#)).
- [34] A. Izmailov e M. Solodov. *Otimização, Volume II - Métodos de Otimalidade*. 3<sup>a</sup> ed. Vol. 1. 2 vol. Rio de Janeiro: IMPA, 2007 (ver pp. [17](#), [18](#)).
- [35] M. Jaggi. “Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization”. In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. por S. Dasgupta e D. McAllester. Vol. 28. Proceedings of Machine Learning Research. Atlanta, Georgia, USA: PMLR, jun. de 2013, pp. 427–435 (ver pp. [11](#), [12](#), [19](#), [20](#), [39](#), [73](#)).
- [36] B. Kalantari e M. Saks. “On the triangle algorithm for the convex hull membership”. In: *23rd annual fall workshop on computational geometry*. City College of New York, 2013 (ver p. [47](#)).
- [37] B. Kalantari. “A Characterization Theorem and an Algorithm for a Convex Hull Problem”. In: *Annals of Operations Research* 226.1 (ago. de 2014), pp. 301–349 (ver pp. [10](#), [11](#), [29](#), [32–38](#)).
- [38] B. Kalantari. “A Triangle Algorithm for Semidefinite Version of Convex Hull Membership Problem”. In: *arXiv:1904.09854 [cs, math]* (mai. de 2019). arXiv: [1904.09854 \[cs, math\]](#) (ver p. [10](#)).
- [39] B. Kalantari. “An Algorithmic Separating Hyperplane Theorem and Its Applications”. In: *Discrete Applied Mathematics*. Distance Geometry Theory and Applications (DGTA 16) 256 (mar. de 2019), pp. 59–82 (ver pp. [10](#), [11](#), [47](#)).
- [40] B. Kalantari. “On the Equivalence of SDP Feasibility and a Convex Hull Relaxation for System of Quadratic Equations”. In: *arXiv:1911.03989 [cs, math]* (ago. de 2020). arXiv: [1911.03989 \[cs, math\]](#) (ver p. [10](#)).

- [41] B. Kalantari e J. Y. Park. “Three Convex Hull Theorems on Triangles and Circles”. *In: Honam Mathematical J.* 36.4 (dez. de 2014), pp. 787–794 (ver p. 10).
- [42] B. Kalantari e Y. Zhang. “Algorithm 1024: Spherical Triangle Algorithm: A Fast Oracle for Convex Hull Membership Queries”. *In: ACM Transactions on Mathematical Software* 48.2 (jun. de 2022), pp. 1–32 (ver pp. 12, 13, 52, 57, 60, 61, 63, 70).
- [43] B. Kalantari e Y. Zhang. “Spherical Triangle Algorithm: A Fast Oracle for Convex Hull Membership Queries”. *In: arXiv:1810.07346 [cs]* (abr. de 2019). arXiv: 1810.07346 [cs] (ver pp. 10, 73).
- [44] Y. Kanno. “An accelerated Uzawa method for application to frictionless contact problem”. *In: Optimization Letters* 14.7 (2020), pp. 1845–1854 (ver p. 68).
- [45] S. Lacoste-Julien e M. Jaggi. “On the global linear convergence of Frank-Wolfe optimization variants”. *In: Proceedings of the 28th International Conference on Neural Information Processing Systems*. Vol. 1. NIPS’15. Cambridge, MA, USA: MIT Press, dez. de 2015, pp. 496–504 (ver pp. 11, 12, 19, 22, 24, 25, 73).
- [46] Y. LeCun, L. Bottou, Y. Bengio e P. Haffner. “Gradient-Based Learning Applied to Document Recognition”. *In: Proceedings of the IEEE* 86.11 (nov. de 1998), pp. 2278–2324 (ver p. 54).
- [47] M. Li e B. Kalantari. “Experimental study of the convex hull decision problem via a new geometric algorithm”. *In: 23rd Annual Fall Workshop on Computational Geometry, City College of New York*. 2013 (ver pp. 12, 29, 46).
- [48] D. G. Luenberger e Y. Ye. *Linear and nonlinear programming*. 3<sup>a</sup> ed. 116. New York, NY: Springer, 2008 (ver pp. 11, 16, 17, 20, 25).
- [49] L. Martein e S. Schaible. “On solving a linear program with one quadratic constraint”. *In: Rivista di matematica per le scienze economiche e sociali* 10.1 (1987), pp. 75–90 (ver pp. 64, 65).
- [50] B. Martinet. “Régularisation d’inéquations variationnelles par approximations successives. Rev. Française Informat”. *In: Recherche Opérationnelle* 4 (1970), pp. 154–158 (ver p. 20).



- [51] J. Peña, D. D. Rodríguez e N. Scheili. “On the von Neumann and Frank–Wolfe Algorithms with Away Steps”. In: *SIAM Journal on Optimization* 26.1 (2016), pp. 499–512 (ver p. 41).
- [52] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970 (ver pp. 14, 17).
- [53] C. Tóth, J. O’Rourke e J. E. Goodman, ed. *Handbook of discrete and computational geometry*. 3<sup>a</sup> ed. Boca Raton: CRC Press, 2017 (ver p. 10).
- [54] R. Yousefzadeh. “Deep Learning Generalization and the Convex Hull of Training Sets”. In: *arXiv:2101.09849 [cs, math]* (jan. de 2021). arXiv: 2101.09849 [cs, math] (ver pp. 54, 55).
- [55] R. Yousefzadeh e F. Huang. “Using Wavelets and Spectral Methods to Study Patterns in Image-Classification Datasets”. In: *arXiv:2006.09879 [cs, eess, math]* (jun. de 2020). arXiv: 2006.09879 [cs, eess, math] (ver p. 55).
- [56] H. Zhang e W. W. Hager. “A Nonmonotone Line Search Technique and Its Application to Unconstrained Optimization”. In: *SIAM Journal on Optimization* 14.4 (jan. de 2004), pp. 1043–1056 (ver pp. 11, 26).
- [57] Y. Zhang e B. Kalantari. “The Triangle Algorithm with Relaxed and Randomized Pivots”. In: *Proceedings of 26th Fall Workshop on Computational Geometry*. url: [https://matthewpjohnson.org/fwcg2016/FWCG\\_2016\\_paper\\_30.pdf](https://matthewpjohnson.org/fwcg2016/FWCG_2016_paper_30.pdf). CUNY Graduate Center. New York, NY, 2016 (ver p. 47).

## APÊNDICE A – DEMONSTRAÇÃO LEMA 3.2

A seguir temos a demonstração do Lema 3.2.

*Demonstração.* Como  $\text{conv}(\mathcal{A})$  é um conjunto não-vazio, fechado, e convexo, pelo Teorema 2.9, se  $p \notin \text{conv}(\mathcal{A})$ , existe um único  $p^+ \in \text{conv}(\mathcal{A})$  tal que

$$\|v - p\| > \|v - p^+\|, \quad \forall v \in \text{conv}(\mathcal{A}) \setminus \{p^+\}.$$

Como todo  $v_j \in \mathcal{A}$  pertence ao  $\text{conv}(\mathcal{A})$ , usando  $p' = p^+$  provamos que  $d(v_j, p) > d(v_j, p'), \forall v_j \in \mathcal{A}$ .

Por outro lado, considere agora que existe  $p' \in \text{conv}(\mathcal{A})$  tal que  $d(v_j, p) > d(v_j, p'), \forall v_j \in S$ , isto é,  $\|v_j - p\| > \|v_j - p'\|, \forall v_j \in \mathcal{A}$ . Em particular,

$$\|v_j - p\| > \|v_j - p'\|, \quad \forall v_j \in E,$$

em que  $E \subset \mathcal{A}$  é o subconjunto de  $\mathcal{A}$  dos pontos extremos de  $\text{conv}(\mathcal{A})$ . Daí, de  $\|v_j - p\|^2 > \|v_j - p'\|^2$ , segue que

$$\frac{\|p\|^2 - \|p'\|^2}{2} > (p - p')^T v_j, \quad \forall v_j \in E. \quad (65)$$

Além disso,  $\forall v \in \text{conv}(\mathcal{A})$ ,  $v$  é combinação convexa dos  $v_j$ 's em  $E$ , isto é,

$$v = \sum_{j=1}^{|E|} \alpha_j v_j, \text{ em que } \sum_{j=1}^{|E|} \alpha_j = 1, \quad \alpha_j \geq 0, \quad \forall j.$$

Então, de (65), temos que  $\forall v \in \text{conv}(\mathcal{A})$

$$(p - p')^T v = (p - p')^T \left( \sum_{j=1}^{|E|} \alpha_j v_j \right) < \frac{\|p\|^2 - \|p'\|^2}{2}. \quad (66)$$

Em particular, para  $v = p'$  temos

$$\frac{\|p\|^2 - \|p'\|^2}{2} > (p - p')^T p',$$

do que segue que

$$\begin{aligned} (p - p')^T p + \frac{\|p\|^2 - \|p'\|^2}{2} &> (p - p')^T p + (p - p')^T p' \\ &= (p - p')^T (p + p') \\ &= \|p\|^2 - \|p'\|^2, \end{aligned}$$

e portanto

$$(p - p')^T p > \frac{\|p\|^2 - \|p'\|^2}{2}. \quad (67)$$

De (66) e (67) concluímos que  $p \notin \text{conv}(\mathcal{A})$ . □

## APÊNDICE B – DEDUÇÃO DAS CONSTANTES DA COMPLEXIDADE DE ITERAÇÃO (TABELA 1)

Considerando o problema (22), no caso que  $p \in \text{conv}(\mathcal{A})$ , temos que  $\Psi(y^*) = 0$ ,  $L = \mu = 1$ . Deduziremos a seguir as constantes que aparecem na complexidade de iteração da Tabela 1.

De (12), para o método de Frank-Wolfe temos

$$\Psi(y_k) = \frac{1}{2} \|y_k - p\|^2 \leq \frac{2D^2}{k+2},$$

sendo assim, para que

$$\sqrt{\frac{4D^2}{k+2}} \leq \varepsilon,$$

é suficiente que

$$\frac{4D^2}{\varepsilon^2} - 2 \leq k.$$

Já para Frank-Wolfe satisfazendo a Hipótese 3.16 conseguimos outras constantes. De (16) e utilizando o fato que

$$1 + t \leq \exp t, \forall t \in \mathbb{R}, \quad (68)$$

temos

$$\begin{aligned} \|y_{k+1} - p\|^2 &\leq \left[1 - \frac{\rho^2}{D^2}\right] \|y_k - p\|^2 \\ &\leq \exp\left[-\frac{\rho^2}{D^2}\right]^k \|y_0 - p\|^2. \end{aligned}$$

Dessa forma, para que

$$\exp\left[-\frac{\rho^2}{D^2}\right]^{k/2} \|y_0 - p\| \leq \varepsilon,$$

é suficiente que

$$\frac{2D^2}{\rho^2} \ln \frac{\delta_0}{\varepsilon} \leq k,$$

em que  $\delta_0 = \|y_0 - p\|$ .

Para finalizar a dedução das constantes da complexidade de iteração (Tabela 1), segue os resultados para o Away Step Frank Wolfe.

Note que de (18), utilizando (68), temos

$$\begin{aligned} \|y_{k+1} - p\|^2 &\leq \left[1 - \frac{1}{4} \left(\frac{\Omega_C}{D(m+1)}\right)^2\right]^{(k-1)/2} \|y_0 - p\|^2 \\ &\leq \exp\left[-\frac{1}{4} \left(\frac{\Omega_C}{D(m+1)}\right)^2\right]^{(k-1)/2} \|y_0 - p\|^2. \end{aligned}$$

Dessa forma, queremos que

$$\exp \left[ -\frac{1}{4} \left( \frac{\Omega_C}{D(m+1)} \right)^2 \right]^{(k-1)/4} \|y_0 - p\| \leq \varepsilon,$$

implicando que

$$\frac{16D^2(m+1)^2}{\Omega_C^2} \ln \frac{\delta_0}{\varepsilon} \leq k.$$

## APÊNDICE C – TABELAS ADICIONAIS DA SEÇÃO 3.6.1

A seguir temos as tabelas contendo a média do número de iterações que cada algoritmo utilizou para cada cenário da Seção 3.6.1.

Tabela 6 – Números de iterações (em média) para o caso em que  $p \in \text{conv}(\mathcal{A})$

$n$	Cenário (a)				Cenário (b)	
	TA	ASFW	GT	SPG	ASFW	SPG
500	2557.3	573.9	662.2	23.7	12	8
1000	1544.8	242.8	247.9	15.9	12.5	8.8
1500	1428.1	195.0	196.5	15.0	13	9.3
2000	1373.0	167.4	169.7	13.8	12	8.0
2500	1317.2	153.4	153.7	13.1	12	7.8
3000	1345.9	147.3	146.1	13.1	12.5	10
3500	1288.7	139.1	139.5	13.1	12	8.6
4000	1295.5	132.3	133.3	13.0	12	8.9
4500	1270.0	129.8	129.8	13.0	13	10.3
5000	1309.7	126.1	126.6	13.0	12	8.4
$1 \times 10^4$	1261.5	110.7	110.5	12.0	13	11.8
$2 \times 10^4$	1254.6	99.0	99.0	11.9	13	12.4
$3 \times 10^4$	1220.8	93.4	93.4	12.0	13	10.6
$4 \times 10^4$	1251.6	88.0	87.8	12.0	12	11.0
$5 \times 10^4$	1235.0	86.7	86.7	12.0	12	12.6
$6 \times 10^4$	1243.1	84.1	84.1	12.0	13	12.8
$7 \times 10^4$	1265.8	83.5	83.4	12.0	12	9.4
$8 \times 10^4$	1247.3	82.2	82.2	12.0	13	13.1
$9 \times 10^4$	1233.6	80.7	80.8	12.0	13	11.0
$10^5$	1225.5	79.3	79.4	12.0	12	10.2

Tabela 7 – Números de iterações (em média) para o caso em que  $p \notin \text{conv}(\mathcal{A})$

$n$	Cenário (c)				Cenário (d)			
	TA	ASFW	GT	SPG	TA	ASFW	GT	SPG
500	2.3	1	1	1.3	6570.6	9.2	6575.2	4.6
1000	3	1	1	1.3	7347.5	9.1	7358.4	4.0
1500	1.9	1	1	1.2	6474	9.2	6483.2	4.8
2000	2.6	1	1	1.3	7233.5	9.0	7246.2	4.5
2500	2.2	1	1	1.5	6604	9.1	6616.4	4.1
3000	2.2	1	1	1.5	6657.9	9.2	6670.2	4.6
3500	3	1	1	1.4	6519.3	9.2	6529.5	4.0
4000	2.0	1	1	1.7	5553.0	9.2	5560.4	4.8
4500	2.7	1	1	1.5	6275.2	9.2	6283.2	5.1
5000	2.3	1	1	1.5	6628.0	9.1	6638.5	4.3
$1 \times 10^4$	3.3	1	1	1.3	6154.4	9.1	6165.8	4.9
$2 \times 10^4$	2.1	1	1	1.8	5671.3	9.2	5680.0	5.2
$3 \times 10^4$	2.6	1	1	1.6	6184.6	9.0	6201.9	4.0
$4 \times 10^4$	2.9	1	1	1.6	6146.2	9.0	6159.5	4.0
$5 \times 10^4$	2.6	1	1	1.6	5795.8	9.2	5811.4	5
$6 \times 10^4$	2.6	1	1	1.6	6390.5	9.1	6416.2	4.4
$7 \times 10^4$	3.4	1	1	1.5	5890.2	9.0	5903.4	4.6
$8 \times 10^4$	2.7	1	1	1.6	5783.6	9.1	5803.0	4.9
$9 \times 10^4$	2.6	1	1	1.4	5324	9.2	5334.4	5
$10^5$	2.6	1	1	1.7	5542.4	9.0	5560.3	4.4