



Universidade Federal de Minas Gerais  
Curso de Graduação em Engenharia Elétrica  
Curso de Graduação em Engenharia De Controle e Automação  
Curso de Graduação em Engenharia de Sistemas  
Departamento de Engenharia Elétrica da UFMG

## **TRABALHO II**

### **OTIMIZAÇÃO EM ENGENHARIA**

**MÉTODO DE PONTOS INTERIORES CONJUGADO COM O SIMPLEX PARA  
A RESOLUÇÃO DO PROBLEMA DE KLEE-MINTY**

Aluno: Giovanni Martins de Sá Júnior – Matrícula 2017001850

Data: 17/07/2024

---

# Sumário

---

<b>Sumário .....</b>	<b>i</b>
<b>Capítulo 1.....</b>	<b>2</b>
1.1. Problemas de Programação Linear.....	2
1.2. Método SIMPLEX.....	3
1.3. Método de Pontos Interiores .....	3
1.4. Método Híbrido – SIMPLEX e PI.....	4
1.5. Problema de Klee-Minty.....	5
<b>Capítulo 2.....</b>	<b>8</b>
<b>Capítulo 3.....</b>	<b>10</b>
3.1. Variação da Dimensão do Espaço.....	10
3.2. Variação do Passo.....	14
3.3. Variação do Gap de Dualidade.....	18
3.4. Análise do Chaveamento entre os Métodos PI e SIMPLEX.....	22
3.5. Ponto Inicial e Parâmetros do Problema de Klee-Minty .....	24
<b>Capítulo 4.....</b>	<b>25</b>
<b>Referências Bibliográficas.....</b>	<b>26</b>

---

# Capítulo 1

## Introdução

---

### 1.1. Problemas de Programação Linear

Os problemas de programação linear são problemas matemáticos de otimização compostos por funções de otimização lineares e contínuas. A programação linear é uma área de estudo considerada pelos pesquisadores extremamente importante, pois essa área do conhecimento proporciona o desenvolvimento de ferramentas computacionais capazes de resolver diversos problemas práticos em pesquisa operacional, que podem ser expressos por meio de funções lineares (LUENBERGER, 2008).

A formulação de um problema de programação linear consiste em determinar as variáveis de decisão, a função objetivo e, no caso de problemas restritos, um conjunto de restrições dadas sob a forma de desigualdades e/ou igualdades. As variáveis de decisão são as incógnitas do problema, ou seja, as variáveis desconhecidas a serem determinadas pelo método de solução. A função objetivo é uma função matemática a ser minimizada ou maximizada que define a qualidade da solução em função das variáveis de decisão. Além disso, o modelo deve levar em conta as limitações físicas do sistema por meio de incluir um conjunto de restrições que limitam as variáveis de decisão a seus valores viáveis (factíveis). As formas padrões de um problema de programação linear podem ser descritas conforme apresentado abaixo (LUENBERGER, 2008):

Primal	Dual	Primal-Dual
minimizar $c^T x$	maximizar $b^T y$	$Ax = b$
sujeito a: $Ax = b$	sujeito a: $A^T y + z = c$	$A^T y + z = c$
$x \geq 0$	$z \geq 0, y \text{ livre}$	$xz \geq 0$
		$x, z \geq 0$

Uma vez formulado o problema a ser otimizado (minimização ou maximização), diversos métodos desenvolvidos para resolução dos problemas de programação linear podem ser aplicados. Entre eles se destacam os métodos SIMPLEX e de Pontos Interiores (PI). Sendo que cada qual possui as suas vantagens e desvantagens. O método de PI possui a característica de convergir rapidamente para a solução dos problemas de otimização, porém, a solução proposta por esse método nunca é igual ao valor do vértice ótimo. Já o método Simplex, diferentemente do método de Pontos Interiores, encontra a solução ótima, mas a sua aplicação em alguns problemas, como por exemplo, o hipercubo m-dimensional de Klee-Minty, pode ser extremamente ineficiente devido à exigência computacional (DEZA, et al., 2006) (SILALAH, 2011).

Dessa forma, busca-se conjugar os dois métodos, Pontos Interiores e SIMPLEX, para ser possível, em um problema como o do hipercubo de Klee-Minty, encontrar uma solução ótima exata (vértice) e de forma eficiente. Nas seções desse capítulo são

apresentadas os modelos propostos pelos métodos de Pontos Interiores e SIMPLEX, bem como a descrição do problema hipercubo m-dimensional de Klee-Minty.

## 1.2. Método SIMPLEX

A proposta geométrica do método SIMPLEX, publicado em 1951 por Dantzig, consiste em caminhar nos vértices do polítopo definido pelo conjunto de restrições. Esses vértices são pontos extremos pertencentes ao conjunto de soluções viáveis. Dessa forma, tendo como ponto inicial um vértice viável, caminha-se de um ponto extremo a outro ponto extremo adjacente até chegar no vértice ótimo que otimiza ao máximo a função objetivo. A Figura 1 ilustra a ideia geométrica proposta pelo método Simplex (LUENBERGER, 2008).

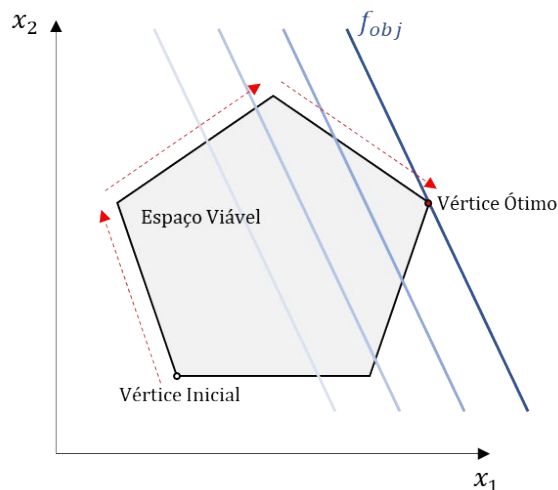


Figura 1 – Visão geométrica do SIMPLEX para um problema de duas dimensões.

O método SIMPLEX proporcionou a resolução eficiente e robusta de diversos problemas práticos, como transporte, produção, alocação de recursos e problemas de escalonamento. Todavia, em 1972, Klee e Minty demonstraram por meio de exemplos, que para determinados problemas de programação linear, o método SIMPLEX precisa avaliar todos os vértices existentes, o que configura a pior condição. Isso requer um número de passos exponencial em relação ao número de variáveis do problema ( $2^n - 1$  iterações, sendo  $n$  o número de variáveis de decisão) (SILALAH, 2011).

Sabe-se que o desenvolvimento dos computadores possibilitou a aplicação do método SIMPLEX a problemas de grande porte, mas, por outro lado, o método revelou alguns dos problemas numéricos que podem ocorrer em cálculos feitos por computadores. Diante desse impasse, muitos pesquisadores buscaram desenvolver algoritmos capazes de solucionar os problemas de programação linear com número de passos polinomial em relação à dimensão do problema. Nesse contexto, a próxima seção é destinada ao entendimento do método de Pontos Interiores.

## 1.3. Método de Pontos Interiores

O método de Pontos Interiores, publicado em 1984 por Karmarkar, gera uma série de pontos no interior da região factível do problema. Esses pontos, chamados de pontos internos viáveis, estão localizados no primeiro octante em que as coordenadas (variáveis de decisão) são estritamente positivas. Dessa forma, define-se uma trajetória central, até obter uma solução próxima do vértice ótimo que satisfaz o critério de erro mínimo admitido entre as soluções do problema primal e dual. Vale ressaltar que o método de Pontos Interiores nunca caminha pela fronteira do conjunto viável e fornece uma solução aproximadamente igual ao vértice ótimo. A Figura 2 ilustra o funcionamento do método de Ponto Interiores do ponto de vista geométrico (LUENBERGER, 2008).

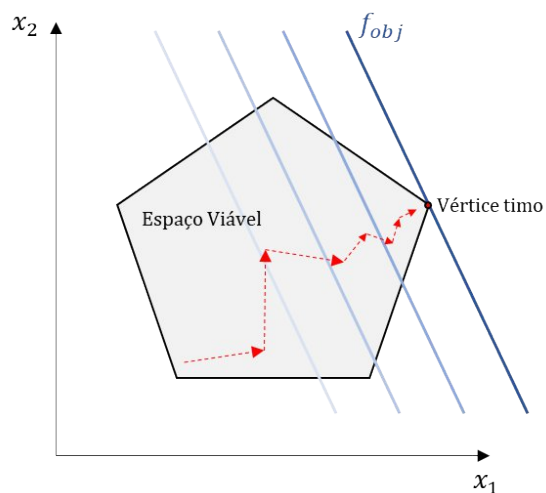


Figura 2 – Visão geométrica do PI para um problema de duas dimensões.

A complexidade do método de Pontos Interiores é polinomial. Dessa forma, apesar da solução determinada por esse método não ser igual ao vértice ótimo, assim como no caso do SIMPLEX, para determinados tipos de problemas, sua convergência é mais rápida que o mesmo.

#### 1.4. Método Híbrido – SIMPLEX e PI

Como os métodos SIMPLEX e de Pontos Interiores apresentam vantagens e desvantagens distintas, diversos algoritmos híbridos que unem as melhores características de ambos foram desenvolvidos a fim de solucionar de maneira eficiente e precisa os problemas de programação linear (LUENBERGER, 2008).

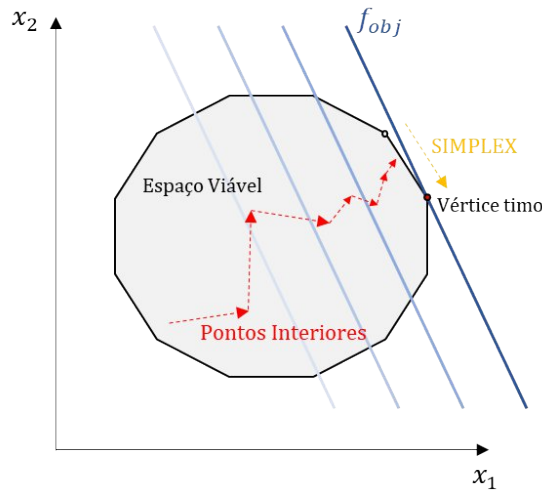


Figura 3 – Visão geométrica da conjugação do PI e SIMPLEX.

A conjugação desses dois métodos se mostra bastante eficiente para solucionar problemas de programação linear complexos, como o apresentado por Klee e Minty, pois por meio dessa conjugação é possível encontrar a solução ótima exata (vértice ótimo do politopo) com baixo custo computacional (com complexidade computacional polinomial).

A rotina de um algoritmo híbrido inicia-se com o método de Pontos Interiores. Este método retorna como solução um valor próximo do vértice ótimo. Após obter uma solução interior aproximada é realizado um chaveamento entre os métodos PI e SIMPLEX. Para tal, o algoritmo passa a buscar um vértice do politopo próximo da solução fornecida pelo método de Pontos Interiores. Um dos algoritmos que realiza essa tarefa é o algoritmo de Murty, o qual realiza uma projeção do ponto interior sobre um hiperplano a fim de buscar um dos vértices do politopo.

O chaveamento entre os métodos envolve uma definição fundamental da otimização linear que consiste na dualidade de representação do politopo por meio de interseções de semiespaço ( $Ax = b$ ) e pelo fecho convexo de um conjunto de pontos. Uma vez que o vértice do politopo, definido através chaveamento, é encontrado, aplica-se o método SIMPLEX a fim de obter a solução exata do problema (vértice ótimo). Dessa forma, o algoritmo efetua uma série de pivoteamentos da solução, caminhando pelos vértices da região factível, até alcançar a solução ótima do problema (MURTY, et al., 1997).

### 1.5. Problema de Klee-Minty

O problema de Klee-Minty a ser resolvido nesse trabalho por meio dos algoritmos SIMPLEX, PI e Híbrido é genericamente descrito conforme apresentado abaixo (SILALAH, 2011):

$$\text{maximize} \quad \sum_{j=1}^n 2^{n-j} \cdot x_j$$

$$\begin{aligned} \text{sujeito a: } & 2 \cdot \sum_{j=1}^{i-1} 2^{i-j} \cdot x_j + x_i \leq valb^{i-1}, & i = 1, 2, \dots, n \\ & x_j \geq 0, & j = 1, 2, \dots, n \end{aligned}$$

Onde  $valb$  é uma constante que determina o vetor de termos independentes do sistema linear formado pelas restrições cuja variação determinada pelos algoritmos fornecidos pelo professor é dada por:  $5 \leq valb \leq 100$ .

Esse trabalho propõe o estudo e aplicação prática dos algoritmos mais utilizados para resolução de problemas de programação linear (método SIMPLEX e de Pontos Interiores). Para tal, realiza-se uma abordagem comparativa dos resultados alcançados pelos algoritmos na resolução de problemas complexos, visando verificar a eficiência (número de iterações e tempo de processamento) e eficácia (erro computacional). A análise é realizada considerando a aplicação dos algoritmos SIMPLEX e PI isoladamente, bem como do algoritmo híbrido dado pelo chaveamento entre os métodos SIMPLEX e PI, via o método de Murty. Por fim, outro estudo que é realizado consiste na análise de sensibilidade da solução quanto a variação dos parâmetros relacionados ao PI (passo e gap de dualidade), do chaveamento entre os dois métodos (SIMPLEX e PI) e da dimensão do espaço (dimensão do problema de Klee-Minty).

Ao considerar  $n = 3$  e  $valb = 100$ , a formulação do problema de otimização linear apresentado acima é descrito como segue (SILALAH, 2011):

$$\begin{aligned} \text{maximize } & 4x_1 + 2x_2 + x_3 \\ \text{sujeito a: } & x_1 \leq 1 \\ & 4x_1 + x_2 \leq 100 \\ & 8x_1 + 4x_2 + x_3 \leq 10000 \\ & x_1, x_2, x_3 \geq 0, \end{aligned}$$

A representação geométrica do problema de três dimensões ( $n = 3$ ) explicitado acima é ilustrada na Figura 4.

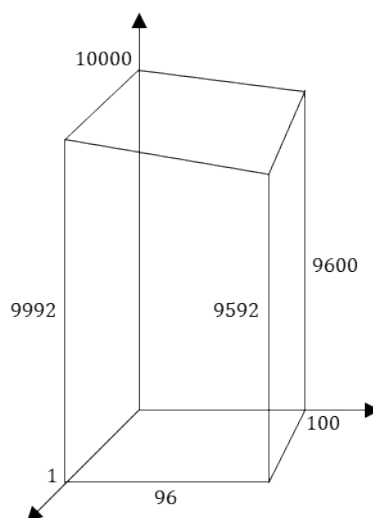


Figura 4 – Hipercubo considerando dimensão três ( $n = 3$ ) do problema de Klee-Minty.

Diante do que foi exposto nesse capítulo, percebe-se que o principal objetivo desse trabalho é a combinação dos métodos SIMPLEX e de Pontos Interiores no intuito de resolver problemas de programação linear complexos, obtendo a solução exata de forma mais eficiente.



---

## Capítulo 2

# Desenvolvimento

---

A primeira etapa para aplicação e análise dos algoritmos que implementam os métodos SIMPLEX, PI e Híbrido é o desenvolvimento de uma rotina capaz de gerar o problema de Klee-Minty. Esse algoritmo possui como parâmetro de entrada a dimensão do espaço e retorna como saída a matriz de restrições ( $A$ ), o vetor de restrição ( $b$ ) e a função a ser maximizada ( $f_{obj}$ ). No caso da dimensão do problema ser igual a três ( $n = 3$ ), tem-se que:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 8 & 4 & 1 \end{bmatrix} \quad b = [1 \quad 100 \quad 10000]^T \quad f_{obj} = [4 \quad 2 \quad 1]$$

A segunda etapa é a definição por parte do usuário de duas variáveis relacionadas com o método de Pontos Interiores: o gap de dualidade e o passo. A primeira consiste na menor diferença aceitável entre o valor da função objetivo do problema primal e o do problema dual descritos na seção 1.1 desse trabalho. Já a segunda equivale ao tamanho da distância percorrida em direção a solução interior ótima (LUENBERGER, 2008). Após a formulação do problema de Klee-Minty e a definição do gap e do passo, tem-se como terceira etapa a aplicação do método de Pontos Interiores a fim de determinar uma solução interior.

O método de Pontos Interiores busca uma solução viável dentro do gap de dualidade existente na implementação. Entretanto, como já abordado, esse método não atinge o vértice solução, ou seja, ele se aproxima da solução ótima, mas não a atinge. Dessa forma, a quarta etapa consiste no chaveamento do método de Pontos Interiores para o método SIMPLEX, pois a partir da solução encontrada pelo PI encontra-se um vértice do problema próximo a essa solução.

Ressalta-se que como as restrições são dadas na forma de  $Ax = b$ , é muito fácil saber os hiperplanos mais próximos em relação a um ponto (solução do método PI). Todavia, é muito custoso e não trivial determinar o vértice mais próximo do ponto (WILLIAMSON, 2011).

Finalmente, a última etapa consiste na aplicação do método SIMPLEX onde ocorre o pivoteamento de forma a adaptar o problema inicial à solução encontrada. A partir de então se passa a caminhar pelos vértices da solução, utilizando as iterações do SIMPLEX até encontrar a solução ótima para o problema de Klee-Minty.

No próximo capítulo é apresentado um roteiro para a obtenção dos resultados obtidos por meio dos experimentos realizados, utilizando os algoritmos disponibilizados e analisando a eficiência e a eficácia das soluções obtidas. Essa análise consiste na variação da dimensão do problema, do passo e do gap de dualidade, a fim de avaliar a

precisão, rapidez e taxa de convergência dos algoritmos, bem como o chaveamento entre os dois métodos.

O detalhamento dos algoritmos implementados em ambiente Matlab® e executados nesse trabalho, a fim de solucionar o problema de Klee-Minty é apresentado na Tabela 1:

Tabela 1 – Etapas dos algoritmos utilizados nesse trabalho e suas respectivas funções.

Etapa	Descrição	Algoritmos		
		SIMPLEX	PI	Híbrido
1	Geração do problema de Klee-Minty	<b>Função:</b> <i>fklemint_pi_splx</i>	<b>Função:</b> <i>fklemint_pi_splx</i>	<b>Função:</b> <i>fklemint_pi_splx</i>
2	Definição do ponto de partida	<b>Vértice inicial:</b> Origem	<b>Ponto interior inicial:</b> Ponto dentro da região viável próximo da Origem	<b>Ponto interior inicial:</b> Ponto dentro da região viável próximo da Origem
3	Aplicar o algoritmo de solução	<b>Função:</b> <i>revised_simplex_01</i>	<b>Função:</b> <i>algpint_01</i>	<b>Funções:</b> <i>algpint_01</i> <i>Murtyalgorithm_01</i> <i>revised_simplex_01</i>
4	Gerar os resultados da execução	Vetor solução ótima Valor da $f_{obj}$ Erro solução ótima Erro $f_{obj}$ Número de iterações Tempo de execução	Vetor solução ótima Valor da $f_{obj}$ Erro solução ótima Erro $f_{obj}$ Número de iterações Tempo de execução	Vetor solução ótima Valor da $f_{obj}$ Erro solução ótima Erro $f_{obj}$ Número de iterações Tempo de execução

Vale ressaltar que todos os resultados devem ser gerados por meio da execução dos algoritmos mais atuais e robustos cujas funções, que implementam os métodos SIMPLEX, PI e Híbrido, são identificadas pelo índice 01, como apresentado nas células destacadas na tabela acima.

---

## Capítulo 3

# Experimentos a Realizar

---

---

## Capítulo 2

## Capítulo 3

Os algoritmos utilizados para obtenção dos resultados que solucionam os problemas de programação linear complexos são: SIMPLEX, Pontos Interiores e Híbrido do SIMPLEX e PI com Chaveamento via Método de Murty.

Ressalta-se todos os resultados de tempo de execução são obtidos por meio dos comandos *tic* e *toc* do Matlab®. Para os resultados, deve-se realizar dez execuções, através das quais serão calculadas as médias do tempo disposto. Isto é efetuado a fim de minimizar a variabilidade do tempo de execução, devido a interferência dada por outros programas rodando em plano de fundo do computador.

### 3.1. Variação da Dimensão do Espaço

Na primeira análise estuda-se os resultados dos algoritmos de solução de problemas de programação linear em relação à variação da dimensão do problema. Nesse caso, os valores dos parâmetros de entrada do método de Pontos Interiores, o gap de dualidade e o passo, devem ser fixados em  $10^{-2}$  e 0,5 respectivamente.

Inicialmente deve-se avaliar o custo computacional de cada um dos algoritmos, considerando problemas de dimensão variando na proporção de quatro em quatro, começando da dimensão  $n = 2$  até 26. Nas tabelas abaixo deve-se apresentar o tempo de processamento e o número de iterações *versus* a dimensão do problema para cada algoritmo analisado.

Tabela 2 – Tempo de execução × Dimensão do problema.

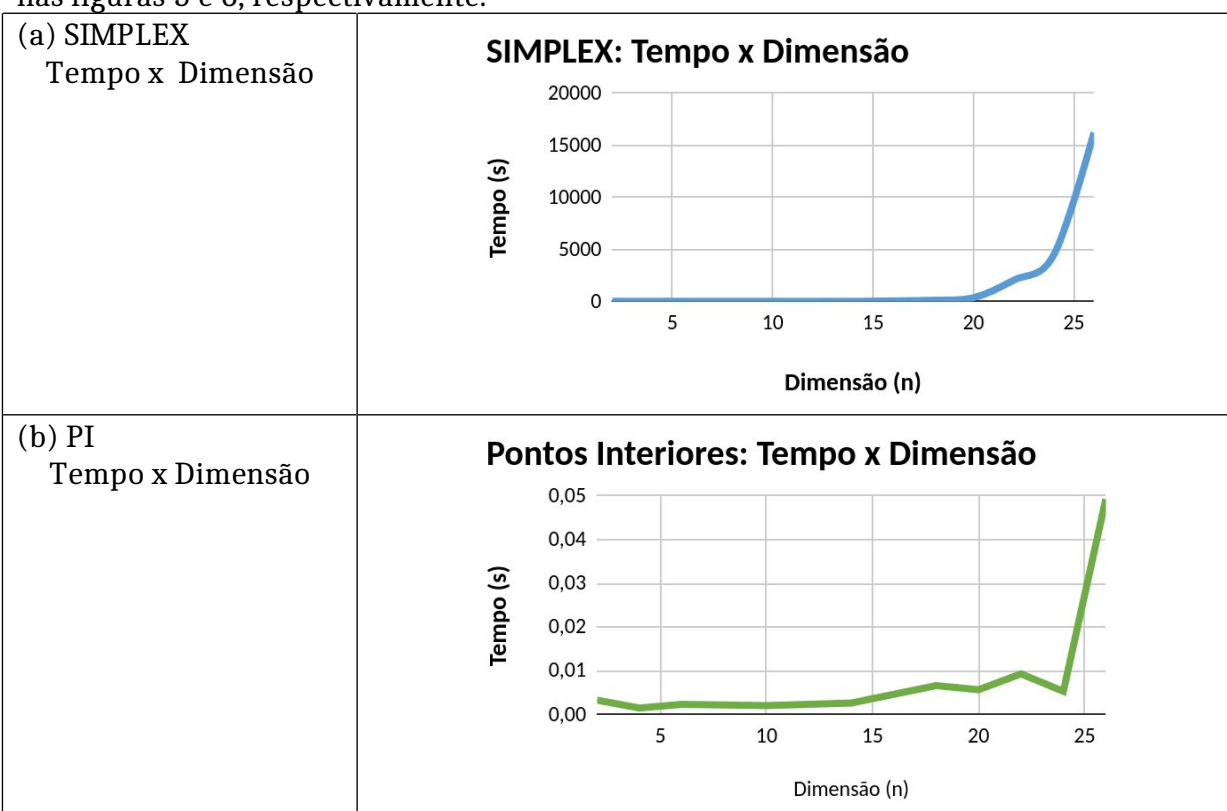
Dimensão	Tempo de execução utilizando a biblioteca Time do Python (s)		
	SIMPLEX	PI	Híbrido
2	0,001429	0,003262	0,002659
4	0,004993	0,001451	0,004326
6	0,008504	0,002289	0,006767
10	0,105543	0,002008	0,043983
14	1,705432	0,002596	0,093210
18	106,0965	0,006580	0,025503
20	342,6043	0,005647	0,020368
22	2027,6066	0,009271	0,009066

24	4460,2306	0,005278	0,008762
26	16223,4563	0,049261	0,015835

Tabela 3 – Número de iterações  $\times$  Dimensão do problema.

Dimensão	Número de iterações do algoritmo		
	SIMPLEX	PI	Híbrido
2	3	7	3
4	15	7	6
6	63	7	15
10	1023	7	207
14	16383	7	360
18	262143	7	8
20	1048575	7	9
22	4194302	7	9
24	16777215	7	9
26	67108863	8	9

Os resultados apresentados nas tabelas 2 e 3 devem ser graficamente ilustrados nas figuras 5 e 6, respectivamente.



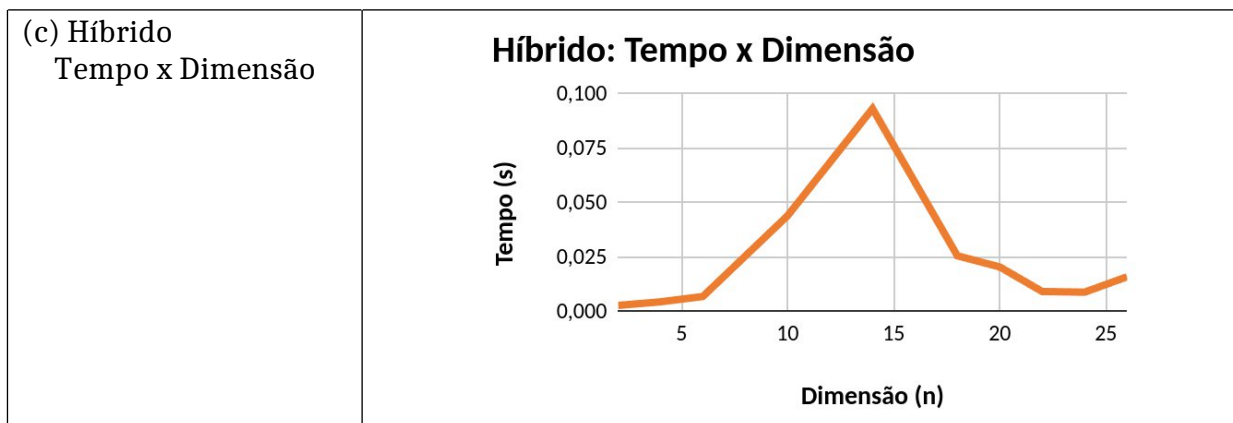


Figura 5 – Custo computacional em função da dimensão do problema sob o ponto de vista do tempo de processamento (a) SIMPLEX, (b) Pontos Interiores, (c) Híbrido.

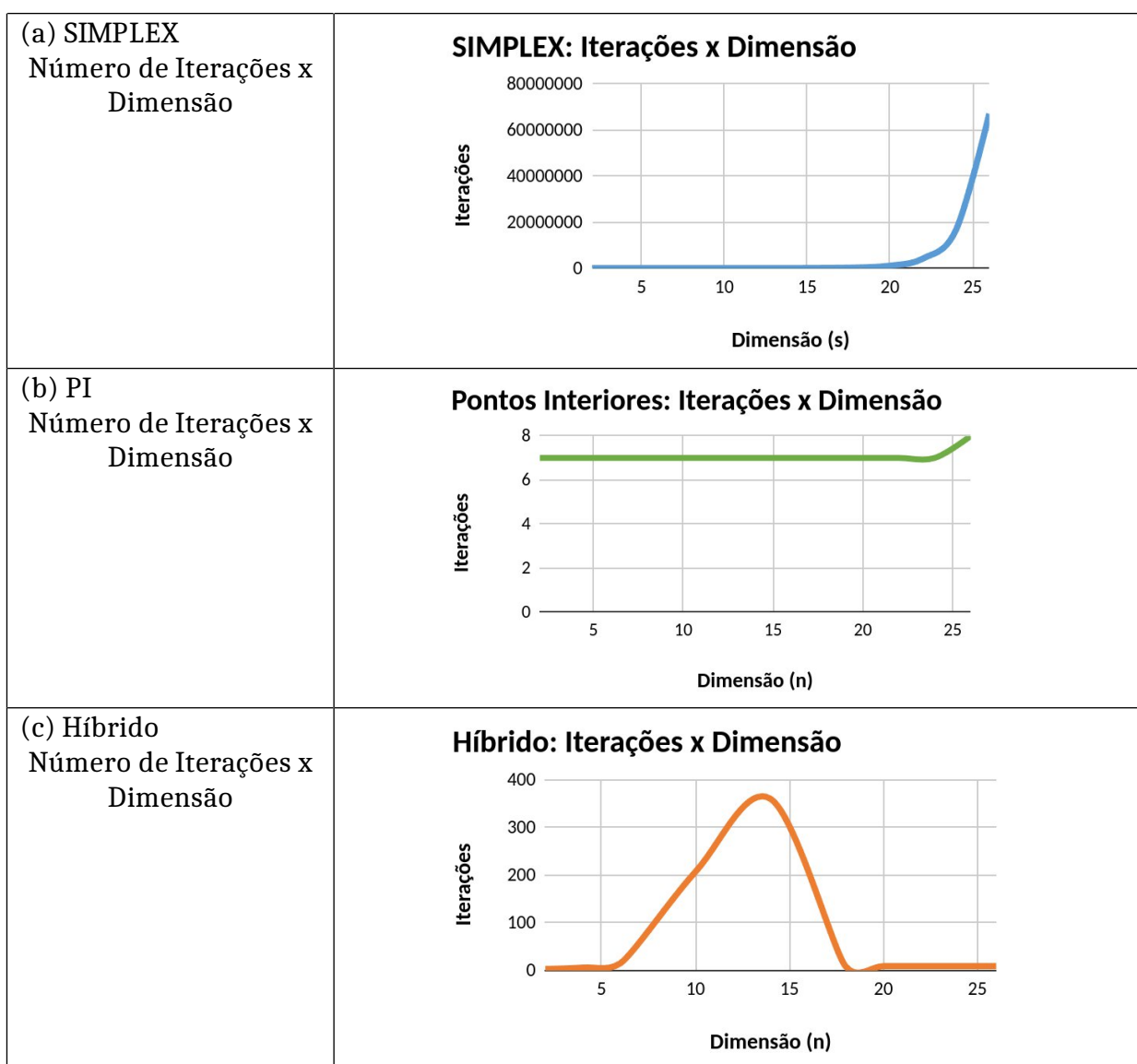


Figura 6 – Custo computacional em função da dimensão do problema sob o ponto de vista do número de iterações (a) SIMPLEX, (b) Pontos Interiores, (c) Híbrido.

A fim de complementar os resultados da análise relacionada à variação da dimensão, deve-se apresentar os resultados de eficácia das soluções. Estes resultados são apresentados nas tabelas abaixo, que informam na ordem em que foram expostas, o erro percentual para o valor final da função objetivo e o erro médio para a solução ótima obtida.

Tabela 4 – Erro percentual da função objetivo  $\times$  Dimensão do problema.

Dimensão	Erro da função objetivo (%)		
	SIMPLEX	PI	Híbrido
2	0	0,8209	2,8571
4	0	0,8210	2,9411
6	0	0,8210	2,9411
10	0	0,8210	2,9411
14	0	0,8210	4,5123
18	0	0,8210	0,8296
20	0	0,8210	0,7812
22	0	0,8210	0,7812
24	0	0,8210	0,7812
26	0	0,6294	0,7812

Tabela 5 – Erro percentual da solução  $\times$  Dimensão do problema.

Dimensão	Erro da solução (%)		
	SIMPLEX	PI	Híbrido
2	0	0,8609	1,4285
4	0	0,8610	1,4287
6	0	0,8610	1,4287
10	0	0,8610	1,4287
14	0	0,8610	2,2610
18	0	0,8610	0,8692
20	0	0,8610	0,7811
22	0	0,8610	0,7813
24	0	0,8610	0,7813
26	0	0,6469	0,7813

- **Realize uma análise completa de todos os resultados obtidos neste capítulo.**

A partir do desenvolvimento desta primeira etapa do trabalho, foi possível observar um comportamento bem interessante da performance dos algoritmos a medida que a dimensão do problema resolvido aumentava gradativamente. Logo de cara, o aumento desta dimensão provocou no algoritmo SIMPLEX, um crescimento exponencial tanto do número de iterações do algoritmo quanto para o tempo necessário para a execução da solução. Como observado pelas figuras 5 e 6, nota-se a curva que ilustra justamente a necessidade do algoritmo de ter visitar um número cada vez maior de vértices do polítopo de uma região viável até que seja encontrada uma solução ótima para o problema. Diante disso, fica justificado o motivo dos erros da solução e da função objetivo deste algoritmo permanecerem zerados.

Em contraposição a este caráter exponencial ao SIMPLEX, o algoritmo dos Pontos Interiores, que é conhecido por ser um algoritmo por promover uma movimentação pelo o interior de uma região viável, demonstrou um comportamento mais interessante frente ao primeiro exemplo, de forma que o número de iterações se manteve basicamente constante a medida que a dimensão do problema aumentava. No problema de maior dimensão ( $N=26$ ), conforme por ser visto na figura 5, o tempo de execução do algoritmo PI passo a apresentar o início de uma tendência de subida, apesar do algoritmo ter conseguido convergir sem maiores custos para o problemas de dimensões menores, apresentando um erro entre 0,8 e 0,9%.

E para o terceiro caso, o algoritmo híbrido, derivado dos dois primeiros também apresentou um comportamento particularmente interessante. Conforme aumentava-se a dimensão do problema, o algoritmo híbrido começou a apresentar uma tendência de crescimento exponencial como o algoritmo SIMPLEX. Tal comportamento foi observado até o momento em que o pico de  $n = 14$ , momento este em que o número de iterações, o tempo de execução e os erros calculados alcançaram um valor limite. Contudo, à medida que a dimensão do problema continuava a aumentar, o algoritmo passou a apresentar um comportamento muito positivo, até melhor que o algoritmo PI.

Assim ficou-se evidenciado que, dependendo da dimensão do problema solucionado o algoritmo de PI pode ser mais interessante de ser utilizado. Contudo, para dimensões muito grandes, o algoritmo híbrido passa a ser um candidato mais interessante.

### 3.2. Variação do Passo

Na avaliação do passo são considerados o algoritmo de Pontos Interiores e o algoritmo híbrido dos métodos SIMPLEX e PI chaveado por meio do método Murty. Nessa análise, o gap de dualidade é fixado em  $10^{-6}$ , enquanto o **passo é variado em valores entre 0 e 1**. O gap de dualidade foi fixado em  $10^{-6}$  para que o seu valor não influencie de forma significativa no estudo da variação do passo. A análise do efeito dessa variação deve considerar os diferentes valores de dimensão do problema ( $n$ ), a fim de avaliar o comportamento dos algoritmos em função da dimensão do problema de Klee-Minty sob o ponto de vista do passo.

Os resultados obtidos para essa análise, considerando apenas o algoritmo de Pontos Interiores, devem ser apresentados nas tabelas abaixo.

Tabela 6 – Tempo de processamento para a análise da variação do passo para diferentes dimensões.

Passo	Tempo de execução utilizando a biblioteca Time do Python (s) – PI						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
0,01	0,069061	0,022226	0,080761	0,124770	0,891512	1,626304	1,611780
0,05	0,010562	0,044014	0,076025	0,139570	0,745062	0,618594	0,408721
0,10	0,031057	0,053598	0,083835	0,068352	0,299302	0,198081	0,221775
0,30	0,016319	0,014397	0,014272	0,021347	0,083688	0,045695	0,121610
0,50	0,013590	0,006496	0,013726	0,008694	0,078998	0,041490	0,054313
0,70	0,014206	0,005799	0,004654	0,007426	0,024756	0,032442	0,022013

0,80	0,009957	0,003363	0,007217	0,005773	0,022121	0,042305	0,076743
0,90	0,003154	0,004525	0,006308	0,008957	0,040318	0,009748	0,017913
0,95	0,008299	0,003737	0,003235	0,004781	0,007104	0,009925	0,039322
0,99	0,002228	0,002780	0,003499	0,009239	0,015118	0,044363	0,013700

Tabela 7 – Número de iterações para a análise da variação do passo para diferentes dimensões.

Passo	Número de iterações do algoritmo – PI						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
0,01	40	120	200	280	360	440	520
0,05	40	120	200	280	284	284	284
0,10	40	120	139	139	139	139	139
0,30	40	42	42	42	42	42	42
0,50	21	22	22	22	22	22	22
0,70	13	14	14	14	14	14	13
0,80	11	11	11	11	11	11	11
0,90	9	9	9	9	9	9	19
0,95	7	8	8	8	8	8	8
0,99	5	6	6	6	6	6	6

Os resultados apresentados nas tabelas 6 e 7, exceto para o valor de passo igual a 0,01, devem ser graficamente ilustrados na Figura 7.

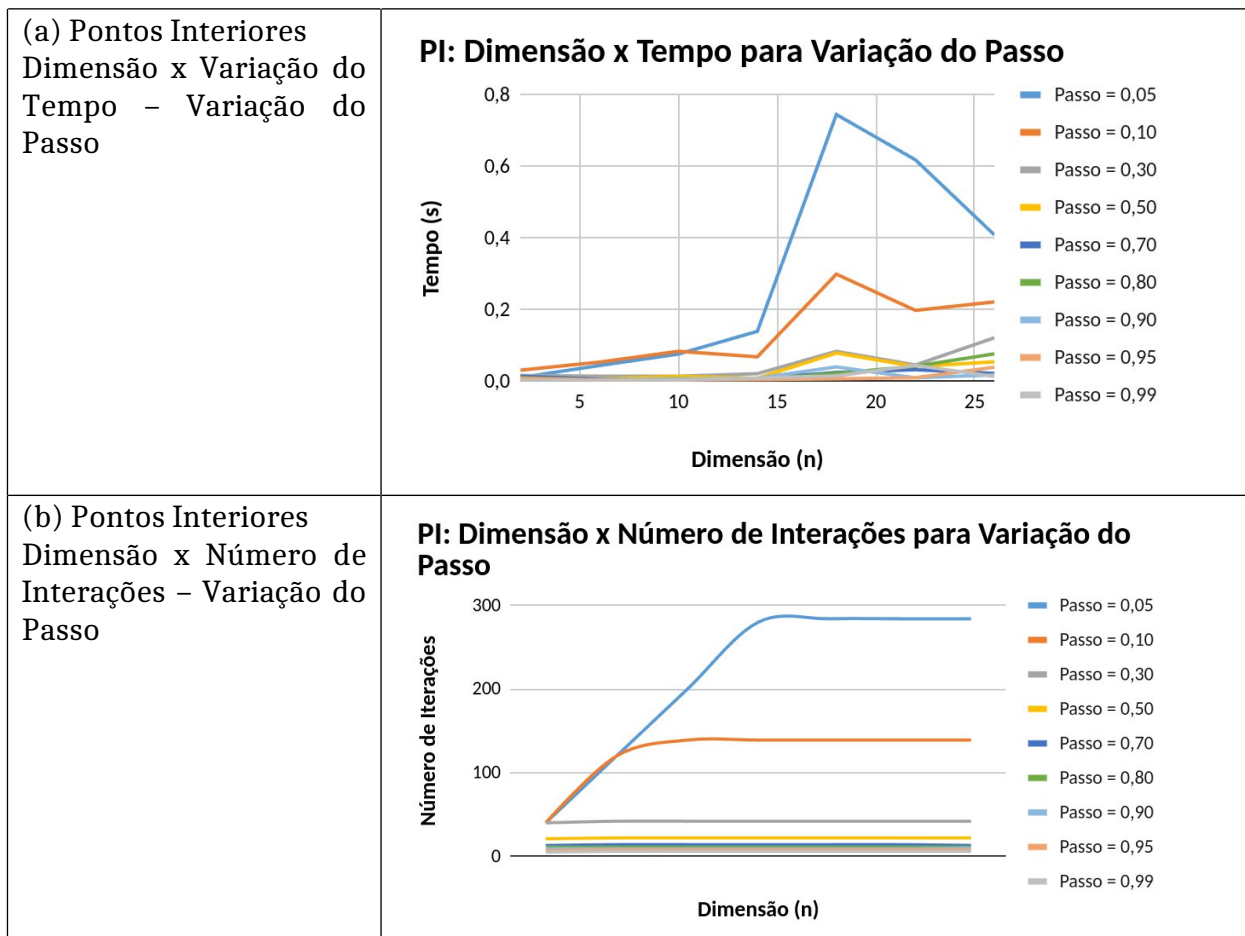


Figura 7 – Análise da sensibilidade do passo (a) Tempo de processamento (b) Número de Interações.



As próximas tabelas devem apresentar os resultados referentes aos erros do valor da função objetivo e da solução obtidos por meio da execução do algoritmo PI para cada dimensão e passo analisado.

Tabela 8 – Erro percentual da função objetivo (Pontos Interiores).

Passo	Erro da função objetivo (%)						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
0,01	66,8979	29,9388	13,3987	5,9967	2,6841	1,2016	0,5408
0,05	12,8552	0,2162	0,0054	0,0001	0,0000970	0,0000970	0,0000970
0,10	1,4862	0,0006	9,30E-05	9,30E-05	9,30E-05	9,30E-05	9,30E-05
0,30	0,0001	7,59E-05	7,59E-05	7,59E-05	7,59E-05	7,59E-05	8,50E-05
0,50	9,50E-05	6,56E-05	6,56E-05	6,56E-05	6,56E-05	6,56E-05	8,66E-05
0,70	7,12E-05	3,70E-05	3,70E-05	3,70E-05	3,70E-05	3,70E-05	9,91E-05
0,80	6,06E-05	9,20E-05	9,20E-05	9,20E-05	9,20E-05	9,20E-05	5,71E-05
0,90	3,85E-05	6,79E-05	6,79E-05	6,79E-05	6,79E-05	6,79E-05	4,01E-05
0,95	9,32E-05	3,57E-05	3,57E-05	3,57E-05	3,57E-05	3,57E-05	7,15E-05
0,99	8,25E-05	2,04E-05	2,04E-05	2,04E-05	2,04E-05	2,04E-05	6,88E-05

Tabela 9 – Erro percentual da solução (Pontos Interiores).

Passo	Erro da solução (%)						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
0,01	66,8988	23,9396	13,3996	5,9975	2,6849	1,2024	0,5411
0,05	12,8593	0,2202	0,0074	0,0001	0,0000970	0,0001000	0,0001000
0,10	1,4943	0,0009	0,0001	0,0001	0,0001	0,0001	0,0001
0,20	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
0,50	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
0,70	0,0001	6,18E-05	6,18E-05	6,18E-05	6,18E-05	6,18E-05	0,0001
0,80	9,93E-05	0,0001	0,0001	0,0001	0,0001	0,0001	9,00E-05
0,90	4,96E-05	0,0001	0,0001	0,0001	0,0001	0,0001	6,32E-05
0,95	0,0001	5,74E-05	5,74E-05	5,74E-05	5,74E-05	5,74E-05	0,0001
0,99	8,93E-05	4,03E-05	4,03E-05	4,03E-05	4,03E-05	4,03E-05	9,79E-05

Tabela 10 – Tempo de processamento para a análise da variação do passo para diferentes dimensões.

Passo	Tempo de execução utilizando a biblioteca Time do Python (s) – Híbrido						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
0,01	0,000888	0,002236	0,219534	0,504395	0,830459	0,962373	1,898194
0,05	0,002834	0,005529	0,089432	0,071304	0,0689784	0,824533	0,771745
0,10	0,004934	0,004839	8,44E-02	1,51E-01	0,344626	0,425361	0,467319
0,30	0,003899	0,003865	0,062395	7,93E-02	0,095334	0,243616	0,194647
0,50	0,002287	0,003609	0,024382	2,94E-02	0,031831	0,086206	0,062057
0,70	0,00186	0,006155	0,026892	3,85E-02	4,66E-02	0,056205	0,099630
0,80	0,003596	0,004321	0,067428	9,43E-03	1,38E-02	0,019283	0,025823
0,90	0,002298	0,009092	0,026267	4,52E-02	6,33E-02	0,019525	0,079037
0,95	0,002889	0,00492	0,030432	3,59E-02	3,78E-02	0,032981	0,028374
0,99	0,002487	0,003923	0,025863	3,59E-02	4,94E-02	0,007045	0,058685

Tabela 11 – Número de iterações para a análise da variação do passo para diferentes dimensões.

Passo	Número de iterações do algoritmo - Híbrido						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
0,01	3	15	207	290	361	441	521
0,05	3	15	207	260	284	272	272
0,10	3	15	207	180	140	134	134
0,30	3	15	207	150	43	42	42
0,50	3	15	207	135	22	23	23
0,70	3	15	207	120	15	14	14
0,80	3	15	207	100	12	11	11
0,90	3	15	207	90	10	9	9
0,95	3	15	207	68	8	7	7
0,99	3	15	207	54	7	6	6

Os resultados apresentados nas tabelas 10 e 11, exceto para o valor de passo igual a 0,01, deve ser graficamente ilustrados na Figura 8.

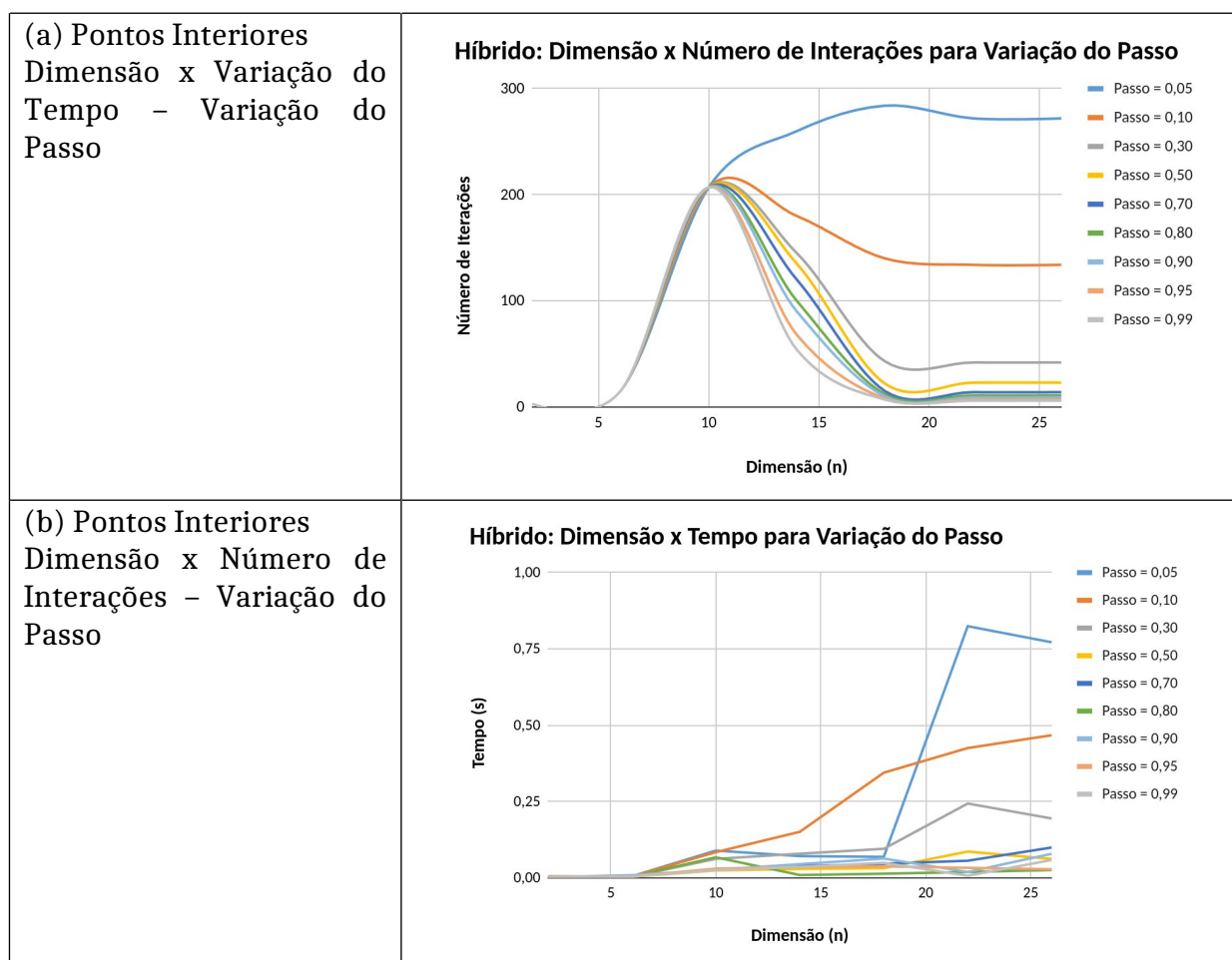


Figura 8 – Análise da sensibilidade do passo (a) Tempo de processamento (b) Número de Iterações.

- **Realize uma análise profunda de todos os resultados.**

Neste segundo momento de discussão do trabalho, será analisado o comportamento dos três algoritmos, tal como no primeiro exemplo, aumentando-se gradativamente a dimensão do problema e também variando-se os passos de zero até um, impactando significativamente na eficiência e precisão dos algoritmos. Contudo, neste caso, a análise ficará restrita ao algoritmo de Pontos Interiores e ao algoritmo híbrido.

Conforme o esperado, o algoritmo de Pontos Interiores sofreu bastante à medida dimensão do problema aumentava. Em contraponto, devido a sua natureza polinomial, à medida em que se aumentava o passo de execução do mesmo, para uma mesma dimensão, o número de iterações necessárias para a resolução do problema caía dramaticamente, assim como o tempo de execução e os erros calculados.

Já para o algoritmo híbrido, o segundo caso apresentou um comportamento bem particular para as diferentes dimensões. Para um passo fixado, o aumento da dimensão provocou na grande maioria dos casos, uma diminuição do número de iterações necessárias para a execução dos problemas, atingindo-se um pico em  $n = 10$ , e convergindo para um menor valor. As únicas duas exceções foram quando os passos foram fixados em valores muito pequenos, como 0,01 e 0,05.

Ademais fixando-se uma dimensão  $n$ , e analisando o comportamento do aumento do passo, em dimensões menores (até  $n = 10$ ), o aumento do passo não influenciou na variação do número de iterações necessárias. Contudo para dimensões maiores (e mais complexas), o aumento do passo favoreceu na diminuição das iterações, tal como visto no algoritmo PI. Junto disso, o tempo de execução também caía à medida que o passo era aumentado, apesar do aumento das dimensões. Os erros encontrados foram muito baixos, variando de ordens de  $10^{-4}$  a  $10^{-5}$ .

### 3.3. Variação do Gap de Dualidade

Na análise dessa seção deve-se avaliar o esforço computacional quanto a variação do gap de dualidade para os algoritmos PI e Híbrido. Nessa análise o passo é mantido em 0,95 (Dica: valor escolhido diante dos estudos apresentados na seção 3.2) e são considerados diferentes valores de dimensão ( $n$ ). Além disso, o gap de dualidade varia entre os valores  $10^{-1}$  e  $10^{-10}$ .

Os resultados para a análise da sensibilidade do gap de dualidade referentes à execução do algoritmo PI devem ser dispostos nas tabelas abaixo.

Tabela 12 – Tempo de processamento para a análise da variação do gap para diferentes dimensões.

Gap de Dualidade	Tempo de execução utilizando os comandos <i>tíc</i> e <i>toc</i> do Matlab (s) – PI						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
$10^{-1}$	0,002308	0,001581	0,001571	0,001782	0,002900	0,013018	0,004761
$10^{-2}$	0,004785	0,002014	0,001434	0,00195	0,005132	0,007171	0,013524
$10^{-4}$	0,003954	0,002339	0,003357	0,002647	0,004151	0,014516	0,012718

$10^{-6}$	0,004801	0,004075	0,006312	0,005512	0,023241	0,012461	0,021247
$10^{-8}$	0,007104	0,006595	0,011245	0,003703	0,035557	0,031788	0,014782
$10^{-10}$	0,004305	0,021982	0,008428	0,014398	0,011286	0,085786	0,05239

Tabela 13 – Número de iterações para a análise da variação do gap para diferentes dimensões.

Gap de Dualidade	Número de iterações do algoritmo – PI						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
$10^{-1}$	2	2	2	2	2	2	3
$10^{-2}$	2	2	2	2	2	2	3
$10^{-4}$	5	5	5	5	5	5	5
$10^{-6}$	7	8	8	8	8	8	8
$10^{-8}$	10	11	11	11	11	11	11
$10^{-10}$	13	14	14	14	14	14	14

Os resultados apresentados nas tabelas 12 e 13, relacionados com o custo computacional do algoritmo PI sob a ótica do gap de dualidade, devem ser graficamente ilustrados na Figura 9.

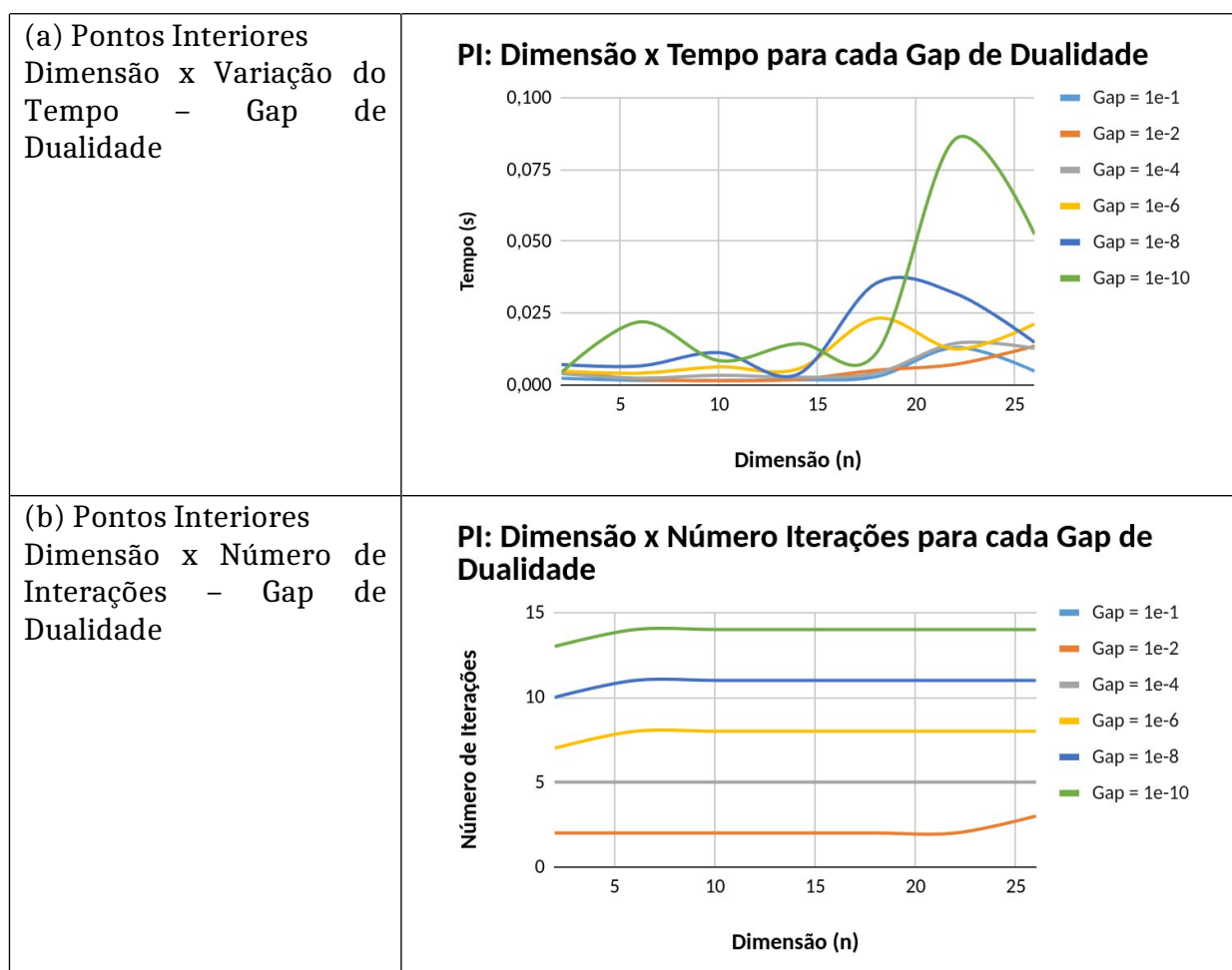


Figura 9 - Análise da sensibilidade do gap (a) Tempo de processamento (b) Número de Iterações.

As próximas tabelas devem apresentar os resultados referentes aos erros do valor da função objetivo e da solução obtidos por meio da execução do algoritmo PI para cada dimensão e gap analisado.

Tabela 14 – Erro percentual da função objetivo (Pontos Interiores).

Passo	Erro da função objetivo (%)						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
$10^{-1}$	0,3262	0,3263	0,3263	0,3263	0,3263	0,3263	0,1803
$10^{-2}$	0,3262	0,3263	0,3263	0,3263	0,3263	0,3263	0,1803
$10^{-4}$	0,0002	0,0002	0,0025	0,0025	0,0025	0,0025	0,006
$10^{-6}$	9,32E-05	3,57E-05	3,57E-05	3,57E-05	3,57E-05	3,57E-05	7,15E-05
$10^{-8}$	6,80E-07	3,64E-07	3,64E-07	3,64E-07	3,64E-07	3,64E-07	9,39E-07
$10^{-10}$	5,00E-09	3,91E-09	3,91E-09	3,91E-09	3,91E-09	3,91E-09	6,74E-09

Tabela 15 – Erro percentual da solução (Pontos Interiores).

Passo	Erro da solução (%)						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
$10^{-1}$	0,4043	0,4045	0,4045	0,4045	0,4045	0,4045	0,2142
$10^{-2}$	0,4043	0,4045	0,4045	0,4045	0,4045	0,4045	0,2142
$10^{-4}$	0,0045	0,0047	0,0047	0,0047	0,0047	0,0047	0,008
$10^{-6}$	0,0001	5,74E-05	5,74E-05	5,74E-05	5,74E-05	5,74E-05	0,0001
$10^{-8}$	8,24E-07	6,07E-07	6,07E-07	6,07E-07	6,07E-07	6,07E-07	1,792E-07
$10^{-10}$	9,209E-09	6,371E-09	6,371E-09	6,371E-09	6,371E-09	6,371E-09	1,045E-09

Por fim, os resultados obtidos para a análise da variação do gap, considerando o algoritmo Híbrido, devem ser apresentados nas tabelas abaixo.

Tabela 16 – Tempo de processamento para a análise da variação do gap para diferentes dimensões.

Gap de Dualidade	Tempo de execução utilizando a biblioteca Time do Python (s) – Híbrido						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
$10^{-1}$	0,002244	0,004629	0,027546	0,003843	0,004563	0,006081	0,037364
$10^{-2}$	0,00447	0,005533	0,039679	0,003271	0,003984	0,004657	0,030204
$10^{-4}$	0,002006	0,007079	0,049098	0,003167	0,006389	0,007775	0,023477
$10^{-6}$	0,023477	0,00498	0,038866	0,003695	0,009421	0,012884	0,016989
$10^{-8}$	0,004051	0,005275	0,028763	0,008424	0,013293	0,039996	0,091209
$10^{-10}$	0,002493	0,004217	0,027673	0,009753	0,018743	0,122595	0,014335

Tabela 17 – Número de iterações para a análise da variação do gap para diferentes dimensões.

Gap de Dualidade	Número de iterações do algoritmo - Híbrido						
	$n = 2$	$n = 6$	$n = 10$	$n = 14$	$n = 18$	$n = 22$	$n = 26$
$10^{-1}$	3	15	207	50	3	4	4
$10^{-2}$	3	15	207	50	5	4	4
$10^{-4}$	3	15	207	51	7	6	6
$10^{-6}$	6	15	207	51	8	7	7
$10^{-8}$	3	15	207	53	11	10	10
$10^{-10}$	3	15	207	53	14	13	13

Os resultados apresentados nas tabelas 16 e 17, relacionados com o custo computacional do algoritmo PI sob a ótica do gap de dualidade, devem ser graficamente ilustrados na Figura 10.

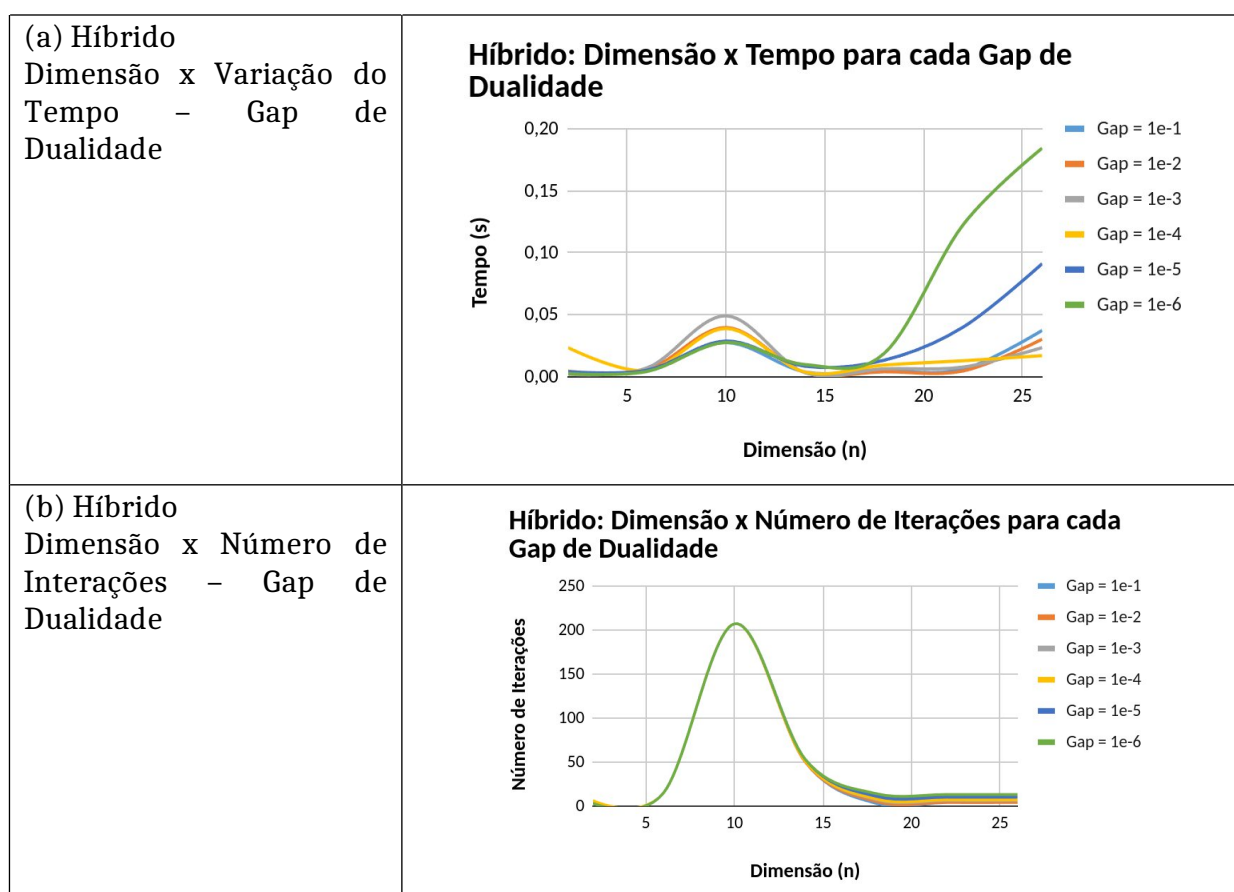


Figura 10 - Análise da sensibilidade do gap (a) Tempo de processamento (b) Número de Iterações.

- **Realize uma análise profunda de todos os resultados obtidos nesta seção.**

Nesta terceira etapa do trabalho, será desenvolvida uma análise do algoritmo de Pontos Interiores e do algoritmo híbrido, sob a perspectiva de analisar o comportamento de ambos no momento em que o passo é delimitado em um valor fixo de 0,95, enquanto o gap de dualidade será gradualmente diminuído, variando de  $10^{-1}$ , até  $10^{-10}$ .

Para o algoritmo de Pontos Interiores, sob a análise de uma mesma dimensão, a diminuição do gap de dualidade provocou um aumento gradual no tempo de execução do algoritmo. Em contrapartida, apesar do aumento, o número de iterações necessárias para cada gap diminuía conforme fosse diminuído o gap de dualidade. Junto disso, o erro da função objetivo e da solução também seguiram o mesmo comportamento da diminuição.

No caso do algoritmo Híbrido, foi observado um comportamento interessante com relação ao tempo gasto em caso de gap. Independente do valor do gap, à medida que a dimensão do problema aumentava, o tempo investido na aumentava, no intervalo de  $0 < n < 10$ , porém, no intervalo de  $10 < n < 16$  (aproximadamente), o tempo gasto em todos os casos convergia pra quase zero. Contudo, a partir de  $n = 16$ , o tempo gasto explodia exponencialmente em praticamente todos os casos.

Com relação ao número de iterações, em basicamente todos os casos de gap, o comportamento dos gráficos se assemelhou a uma curva gaussiana, atingindo o pico em  $n = 10$ , porém convergindo para um valor próximo em dimensões maiores, apesar do aumento do tempo gasto, conforme comentado anteriormente. A grande maioria dos erros calculados se aproximou de zero na maior parte dos casos, contudo, a diminuição do gap de dualidade (como no caso de  $10^{-10}$ ), promoveu uma diminuição ainda maior nos resultados medidos, como foi o caso medido no erro da função objetivo.

### **3.4. Análise do Chaveamento entre os Métodos PI e SIMPLEX**

A grande questão relacionada ao algoritmo Híbrido é quando se deve chavear do algoritmo PI para o SIMPLEX. No intuito de analisar esse chaveamento deve-se realizar um melhor detalhamento quanto aos resultados relacionados ao esforço computacional, considerando as soluções obtidas antes e depois do chaveamento via método de Murty. Dessa forma, os experimentos devem considerar a execução do algoritmo Híbrido para diferentes dimensões, gap de dualidade igual a  $10^{-6}$  e passo igual a 0,95. Os resultados desses experimentos serão dispostos na Tabela 18.

Tabela 18 – Resultados relacionados ao esforço computacional para o algoritmo Híbrido (Detalhamento dos métodos de Pontos Interiores e SIMPLEX).

Dimensão	Iterações		Tempo (s)		Erro $f_{obj}$ (%)		Erro Solução (%)	
	PI	SIMPLEX	PI	SIMPLEX	PI	SIMPLEX	PI	SIMPLEX
2	2	1	0,0013	0,001117	2,8571	0	1,4285	0
4	2	4	0,0011	0,001148	2,9411	0	1,4287	0
6	2	13	0,0012	0,0021	2,9411	0	1,4287	0
10	2	205	0,0009	0,01972	2,9411	0	1,4287	0
14	2	220	0,0239	0,02184	2,9411	0	1,4287	0
18	6	70	0,0184	0,008547	9,94E-05	0	0,0001	0
20	6	1	0,0115	0,003416	9,94E-05	0	0,0001	0
22	6	1	0,0035	0,006984	8,82E-05	0	0,0001	0
24	6	1	0,0031	0,022613	8,82E-05	0	0,0001	0
26	6	1	0,0060	0,004799	8,82E-05	0	0,0001	0

- Realize uma análise aprofundada dos resultados obtidos.

Neste quarto caso, será avaliado a influência dos algoritmos de Pontos Interiores e Simplex, em um contexto de gap de dualidade e passo fixos. A partir dos resultados obtidos é possível afirmar que com relação ao número de iterações, o algoritmo SIMPLEX tende a apresentar um caráter quase que exponencial até o momento de  $n = 14$ . A partir disso, o número de iterações necessárias converge para um nos casos seguintes. Já o algoritmo PI se mantém basicamente constante, sem com que a dimensão interfira de maneira considerável. Acompanhando o comportamento do número de iterações, o tempo requisitado para a resolução do problema seguiu basicamente o mesmo padrão, especialmente no algoritmo SIMPLEX, enquanto o algoritmo PI apresentou um comportamento mais uniforme.

No caso dos erros, como era de se esperar, o algoritmo SIMPLEX apresentou um comportamento zerado, uma vez que o algoritmo é conhecido por apresentar uma alta precisão na solução da função objetivo. Em contrapartida o, algoritmo PI apresentou o comportamento de erro um pouco maior, porém à medida que a dimensão do problema aumentava, a precisão da solução aumentava, fazendo com que o erro calculado diminuísse, sobretudo a partir de  $n = 18$ .



### 3.5. Ponto Inicial e Parâmetros do Problema de Klee-Minty

- **Faça uma análise do comportamento de todos os métodos considerados neste trabalho quanto o ponto inicial e a sensibilidade da constante  $valb$  do problema de Klee-Minty.**

O ponto inicial e a variável da constante  $valb$  são dois parâmetros fundamentais, e influencia diretamente na qualidade da solução proposta dos algoritmos estudados até então.

Com relação ao primeiro parâmetro, o ponto inicial pode afetar consideravelmente na resolução do problema, sobretudo nos algoritmos SIMPLEX e Híbrido (também derivado do SIMPLEX), uma vez que por se moverem apenas através dos vértices, dependendo do ponto inicial escolhido, os algoritmos podem ter que escolher caminhos mais complicados para a resolução do problema proposto. Em contrapartida, o algoritmo de Pontos Interiores não é tão influenciado pelo ponto inicial. Ainda que o ponto inicial escolhido fique longe da solução ótima, o algoritmo de PI tende a convergir melhor. Com isso, durante a utilização dos três algoritmos o ponto inicial escolhido tende a ser mais crítico a medida que o número de dimensões do problema aumenta.

Já o parâmetro  $valb$  desempenha também um papel muito importante no problema de Klee Minty, tanto na definição da complexidade do problema, quanto no comportamento dos algoritmos. Assim, um valor maior para a definição de  $valb$  tende a fazer com que o problema a ser resolvido seja mais complexo, tornando o problema mais custoso aos algoritmos.

---

## Capítulo 4

# Conclusão

---

De acordo com os resultados das sessões anteriores concluir, com base nos experimentos sobre a precisão, a rapidez e a taxa convergência dos métodos de pontos interiores e método SIMPLEX e o híbrido dos dois (observando o que a variação da dimensão do problema pode causar de impacto nos métodos). Finalmente, vocês devem apresentar conclusões gerais sobre o que foi observado nos experimentos.

---

Por meio do desenvolvimento deste trabalho, foi observar que o problema de Klee-Minty pode apresentar diferentes perspectivas na capacidade de resolução do problema, por parte dos algoritmos estudados, principalmente ao se observar a complexidade e eficiência individualmente. Assim, a análise dos três algoritmos sob diferentes perspectivas trouxe detalhes muito importantes sobretudo quando se resolvia problemas com grande dimensão de complexidade.

No algoritmo Simplex, foi possível observar a sua precisão e eficiência, sobretudo em problemas de complexidade pequena e média, encontrando soluções ótimas e com erros praticamente zerados. Em contrapartida, se mostravam sensíveis dependendo do ponto inicial escolhido ou da dimensão do problema.

No algoritmo de Pontos Interiores, eles apresentaram um comportamento mais robusto em problemas de dimensões maiores, apresentando uma maior equilíbrio entre a solução apresentada e os erros calculados. Já o algoritmo híbrido, buscou combinar o benefício de ambos os algoritmos, alternando os métodos conforme o necessário para otimizar a convergência.

Com isso, foi possível observar que não existe uma solução considerada a melhor possível em caso, sendo fundamental realizar uma análise criteriosa do problema proposto e da dimensão a ser resolvida. Diante disso, para problemas mais simples e de menor complexidade, o algoritmo SIMPLEX foi considerada o potencial candidato para a definição da solução ótima. Em contrapartida, para problemas de maior complexidade, o algoritmo Híbrido e o SIMPLEX foram melhores candidatos para a resolução dos problemas.

---

---

## Referências Bibliográficas

---

**DEZA, A., NEMATOLLAHI, E. e TERLAKY, T. 2006.** How Good are Interior Point Methods? Klee–Minty Cubes Tighten Iteration-Complexity Bounds. 2006, pp. 1-14.

**DUNBAR, S. R. 2011.** Probability Theory and Stochastic Processes. *Worst Case and Average Case Behavior of the Simplex Algorithm*. Lincoln, Nebraska, USA : Universidade de Nebraska, 2011.

**LUENBERGER, D. G. 2008.** *Linear and Nonlinear Programming*. 3a. ed. Stanford : Springer Science Business Media, 2008.

**MURTY, K. G. e YU, F. T. 1997.** *Linear Complementarity Linear and Nonlinear Programming*. Internet ed. s.l. : Universidade de Michigan, 1997.

**SILALAH, B. P. 2011.** On the Central Path Redundant Klee-Minty Problems. s.l., Binjai, Indonesia : Tese de Doutorado Submetida à Universidade Técnica de Delf, 2011.

**WILLIAMSON, D. P. 2011.** *New Linear Programming Algorithms and Some Open Problems*. 3a ed. s.l. : Cambridge, 2011, pp. 474-477.