

# Estudo Poliedral do Problema da Variabilidade do Tempo de Resposta

Vicente Avelino de Oliveira Neto



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2019

Vicente Avelino de Oliveira Neto

# Estudo Poliedral do Problema da Variabilidade do Tempo de Resposta

Monografia apresentada ao curso Ciência da Computação do Centro de Informática, da Universidade Federal da Paraíba, como requisito para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Prof. Dr. Gilberto Farias de Sousa Filho

Outubro de 2019

**Catálogo na publicação**  
**Seção de Catalogação e Classificação**

N469e Neto, Vicente Avelino de Oliveira.

Estudo Poliedral do Problema da Variabilidade do Tempo de Resposta / Vicente Avelino de Oliveira Neto. - João Pessoa, 2019.

32 f. : il.

Orientação: Gilberto Farias de Sousa Filho.

Coorientação: Bruno Jefferson de Sousa Pessoa.

Monografia (Graduação) - UFPB/CI.


1. Problema da Variabilidade do Tempo de Resposta. 2. Combinatória Poliédrica. 3. Programação Linear Inteira. I. Filho, Gilberto Farias de Sousa. II. Pessoa, Bruno Jefferson de Sousa. III. Título.

UFPB/CI



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência da Computação intitulado ***Estudo Poliedral do Problema da Variabilidade do Tempo de Resposta*** de autoria de Vicente Avelino de Oliveira Neto, aprovada pela banca examinadora constituída pelos seguintes professores:



Prof. Dr. Gilberto Farias de Sousa Filho

Universidade Federal da Paraíba



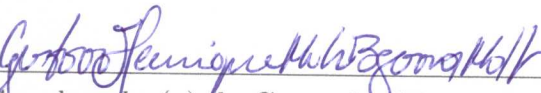
Prof. Dr. Bruno Jefferson de Sousa Pessoa

Universidade Federal da Paraíba



Prof. Dr. Bruno Petrato Bruck

Universidade Federal da Paraíba



Coordenador(a) do Curso de Ciência da Computação

Gustavo Henrique Matos Bezerra Motta

CI/UFPB  
Prof. Dr. Gustavo H. M. B. Motta  
Coordenador Bacharelado  
Ciência da Computação  
CI-UFPB / Mat. SIAPE 2126491

João Pessoa, 27 de setembro de 2019

*“Não esqueças que, às vezes, faz-nos  
falta ter ao lado caras sorridentes.”*

*São Josemaría Escrivá*

## **DEDICATÓRIA**

Aos meus queridos pais, José Avelino e Josinalva Nascimento.

## AGRADECIMENTOS

Agradecimento a Deus pelo Seu amor imenso, Sua infinita bondade e Sua divina misericórdia. Tudo que consegui foi por meio de Sua graça.

À Nossa Senhora, Maria Santíssima, por sempre me guiar e rogar por mim. Tenho a honra de poder chamá-la de Mãe.

Aos meus pais, Seu Avelino e Dona Jó, pelo amor incondicional. Dá para tocar o amor que vocês sentem por mim. Obrigado por fazerem de tudo por mim, me apoiando, me incentivando, me tornando um homem melhor. Eu os amo demais.

Aos meus irmãos, Vinicius e Vitor, por simplesmente existirem em minha vida. O que seria de mim sem a felicidade que vocês me dão?

À Fernanda Patrícia, por ser companheira, paciente, dedicada, carinhosa e teimosa. Você tem noção de todo o bem que me faz, menina?! Tu és muito especial. Ao seu lado só há amor.

Aos meus amigos Caio Eduardo e Luiz Fernando, que facilitaram minha vida pessoal e acadêmica, compartilhando momentos inesquecíveis, sejam eles durante um estudo na noite anterior da prova ou as nossas simples conversas. Gosto muito de vocês.

Aos outros amigos que fiz durante o período universitário: Davysson Silva, Ingrid Almeida, panelinha completa do IF. Os churrascos são sempre bons, mas sempre fica faltando Davysson.

A todos que estiveram comigo no PET Computação. Em especial ao professor Leonardo por toda formação acadêmica e amadurecimento pessoal, e também devo muito aos meus amigos Aline Araújo, Matheus Praxedes, Marcos Alves, Luan Barbosa e Lucca Peregrino, por todos os momentos de aprendizado, companhia, cumplicidade, amizade e sorrisos juntos.

A todos os familiares e amigos que torcem por mim e vibram com as minhas conquistas.

Ao Professor Gilberto Farias pela orientação em trabalhos anteriores e nesta monografia, pelo aprendizado e pela sua imensa paciência e confiança. Agradeço bastante.

Aos professores Bruno Jefferson, Teobaldo Leite e tantos outros que me ajudaram na minha formação, nos trabalhos e me motivaram no curso.

A todos da Universidade Federal da Paraíba pela colaboração e dedicação.

## RESUMO

Seja dado um conjunto de cópias de símbolos e um conjunto de slots, o Problema da Variabilidade do Tempo de Resposta (RTVP) é NP-difícil e objetiva minimizar a variabilidade do tempo de resposta total, atribuindo para cada slot uma e somente uma cópia de símbolo e, juntamente, atribuir cada símbolo em um e somente um slot. Este problema possui um grande número de aplicações, que variam desde a produção de automóveis em uma linha de montagem à coleta de resíduos de recipientes de lixo colocados em várias salas de um hospital. Este trabalho apresenta um estudo poliédrico associado ao RTVP, descrevendo-o em equações e inequações lineares. Nessa perspectiva, realizou-se pesquisa descritiva, quanto aos objetivos, com procedimentos bibliográficos. Além disso, adaptou a descrição em uma formulação matemática baseada em Programação Linear Inteira e comparou-o com um método exato já consolidado na literatura, caracterizando assim como pesquisa de natureza aplicada e abordagem quantitativa. Os resultados computacionais mostram que o modelo proposto sucedeu-se bem, destacando a possibilidade de que as inequações realmente são facetas do politopo.

**Palavras-chave:** Problema da Variabilidade do Tempo de Resposta. Combinatória poliédrica. Programação Linear Inteira.



## ABSTRACT

Given a set of copy of symbols and a set of slots, the Response Time Variability Problem (RTVP) is NP-hard and attempts to minimize the variability of the total response time, attributing to each slot one and only one copy of the symbol and, along, attributing each symbol in one and only one slot. This problem has a great number of applications, varying from automobile production on an assembly line to the collect of residues from a recipient in many hospital rooms. This work presents a polyhedral study associated to RTVP, describing it in linear equations and inequations. From this perspective, a descriptive research was done, in relation to the objectives, with bibliographic procedures. Furthermore, a description in a mathematical formulation was adapted based on Integer Linear Programming and it was compared to an exact method already consolidated on the literature, thus characterizing it as a research of applied nature to a quantitative approach. The computational results show that the proposed model worked well, highlighting the possibility that the inequalities really are polytope facets.

**Key-words:** Response Time Variability Problem. Polyhedral Combinatorics. Integer Linear Programming.

## LISTA DE FIGURAS

- 1      (a) Sequências de símbolos  $S_1$  e  $S_2$  e (b) vetores de incidência  $\chi^{S_1}$  e  $\chi^{S_2}$ .      .      24

## LISTA DE TABELAS

1	Resultados Computacionais . . . . .	30
---	-------------------------------------	----

## **LISTA DE ABREVIATURAS**

AI – Afim Independente

B&B – Branch-and-Bound

GRASP – Procedimento de Busca Adaptativa Gulosa e Randômica

LI – Linearmente Independente

PL – Programação Linear

PLI – Programação Linear Inteira

PLIM – Programação Linear Inteira Mista

PLIP – Programação Linear Inteira Pura

POC – Problema de Otimização Combinatória

PSS – Politopo de Sequenciamento de Símbolos

RTVP – Problema da Variabilidade do Tempo de Resposta

## Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Contextualização . . . . .	13
1.2	Definição do Problema . . . . .	13
1.3	Justificativa . . . . .	14
1.4	Objetivos . . . . .	14
1.4.1	Objetivo geral . . . . .	14
1.4.2	Objetivos específicos . . . . .	14
1.5	Estrutura da monografia . . . . .	15
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>16</b>
2.1	Programação Linear Inteira . . . . .	16
2.2	Combinatória Polédrica . . . . .	17
2.3	Trabalhos Relacionados . . . . .	20
2.3.1	Estudo Poliedral . . . . .	20
2.3.2	Problema da Variabilidade do Tempo de Resposta . . . . .	21
<b>3</b>	<b>METODOLOGIA</b>	<b>23</b>
3.1	Tipologia de Pesquisa . . . . .	23
3.2	Politopo de Sequenciamento de Símbolos . . . . .	23
3.3	Equações lineares de $P_d$ . . . . .	24
3.4	Inequações lineares de $P_d$ . . . . .	27
3.5	Proposta de Modelo . . . . .	28
<b>4</b>	<b>APRESENTAÇÃO E ANÁLISE DOS RESULTADOS</b>	<b>29</b>
<b>5</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>31</b>
	<b>REFERÊNCIAS</b>	<b>31</b>

# 1 INTRODUÇÃO

Este capítulo aborda a contextualização, a definição do problema, a justificativa que evidencia a relevância do presente estudo, bem como os objetivos geral e específicos, e, por fim, a estrutura da monografia.

## 1.1 Contextualização

Os estudos em relação aos Problemas de sequenciamento e escalonamento intensificaram-se a partir da década de 1950, crescendo sua importância em conjunto com o desenvolvimento da indústria e de seus processos de fabricação. Como resultado, tornaram-se uma das mais importantes classes de problemas abordados na área de Pesquisa Operacional. Por causa da notoriedade e do grande número de aplicações que surgiram ao longo do tempo, a literatura que aborda problemas de sequenciamento e escalonamento é vasta e ainda se encontra em crescimento [1].

Para [2], tais problemas lidam com a alocação de tarefas e/ou recursos ao longo do tempo. Tem-se como recursos, por exemplo, slots de tempo em um canal de transmissão, tripulação de empresas aéreas, unidades de processamento em um ambiente de computação multiprocessada, etc. Por outro lado, as tarefas podem ser as execuções dos programas de computador, a transmissão de pacotes de dados em uma rede de computadores, ou também voos comerciais de empresas aéreas [1]. Para fins de simplicidade, daqui em diante os termos tarefas e recursos são tratados apenas como símbolos.

Dentre os problemas existentes, o presente trabalho foca no problema da variabilidade do tempo de resposta (RTVP, do inglês response time variability problem), que consiste em, dado um conjunto de cópias de símbolos e um conjunto de slots, atribuir para cada slot uma e somente uma cópia de símbolo e, simultaneamente, atribuir para cada símbolo um e somente um slot.

## 1.2 Definição do Problema

O RTVP define-se formalmente como: dados  $n$  inteiros positivos  $m_1 \leq \dots \leq m_n$ , defina  $M = \sum_{i=1}^n m_i$  e a taxa  $r_i = \frac{m_i}{M}$  para  $i = 1, \dots, n$ . Considere uma sequência  $S = s_1 s_2 \dots s_M$  de tamanho  $M$ , na qual  $c_i$  (símbolos) ocorre exatamente  $m_i$  vezes (número de cópias para o símbolo  $c_i$ ). A distância  $d$  entre duas cópias consecutivas de  $c_i$  é definida como o número de posições entre elas mais um, já que todos os elementos da sequência possuem o mesmo tamanho.

Desse modo, se  $c_i$  ocorre  $m_i$  vezes, existem exatamente  $m_i$  distâncias  $d_1^i, \dots, d_{m_i}^i$  para  $c_i$ , sendo que  $d_{m_i}^i$  é a distância da última cópia de  $c_i$  de um ciclo para a primeira

cópia de  $c_i$  do ciclo subsequente. O objetivo é minimizar a variabilidade do tempo de resposta total, definida como:

$$RTV = \sum_{i=1}^n \sum_{j \leq m_i} (d_j^i - \bar{d}_i)^2, \quad \text{onde} \quad \bar{d}_i = \frac{M}{m_i} = \frac{1}{r_i}$$

O RTVP trata de escalonamento cíclico com horizonte de tempo finito. O tamanho das sequências é estabelecido a priori a partir das frequências de cada tarefa, que são entradas para os problemas. Corominas et al. [3] definiu o problema e estudou sua complexidade computacional, apresentando uma prova de que o problema é NP-difícil.

### 1.3 Justificativa

Esse problema possui diversas aplicações reais, como, por exemplo, o modelo de produção *Just-in-Time*, que constitui-se em produzir cópias de um mesmo modelo uniformemente quanto possível ao longo do tempo. Consequentemente, a escassez, os estoques excessivos e a ociosidade de máquinas durante o processo de produção reduzem-se [4].

Outra aplicação real foi a descrita por [5], na qual uma unidade de saúde precisava escalonar a coleta de resíduos de recipientes de lixo colocados em várias salas em toda a unidade. Cada sala exigia um número diferente de visitas por turno e cada funcionário tinha uma frequência também diferente para visitar as salas. O objetivo era que essas visitas fossem o mais regular possível para evitar o acúmulo excessivo de lixo em qualquer recipiente. Além disso, não foi encontrado nenhum trabalho que apresente um estudo poliédrico do RTVP. Portanto, acredita-se que o presente estudo desenvolvido é justificável.

### 1.4 Objetivos

#### 1.4.1 Objetivo geral

Este trabalho tem como objetivo apresentar e compreender o poliedro do problema da variabilidade do tempo de resposta.

#### 1.4.2 Objetivos específicos

Os objetivos específicos do trabalho estão listados abaixo:

- Revisar a literatura sobre o problema;

- Modelar o politopo de sequenciamento de símbolos;
- Resolver o problema da variabilidade do tempo de resposta usando o politopo;
- Comparar os resultados em relação ao método exato já implementado da literatura.

## **1.5 Estrutura da monografia**

O restante do trabalho está organizado da seguinte forma: o Capítulo 2 expõe toda fundamentação teórica usada neste trabalho, trazendo conceitos gerais de Programação Linear Inteira e Combinatória Poliédrica, e uma revisão da literatura sobre os trabalhos relacionados ao estudo poliedral e ao RTVP; o Capítulo 3 apresenta a metodologia adotada para o trabalho; o Capítulo 4 mostra e discute os resultados dos experimentos computacionais obtidos; e o Capítulo 5 conclui o trabalho, trazendo considerações e propostas de continuidade.



## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo trata sobre definições e algoritmos da Programação Linear Inteira, conceitos envolvendo a Combinatória Poliédrica, e os estudos anteriores que abordaram sobre o estudo poliedral e o RTVP.

### 2.1 Programação Linear Inteira

A Programação Linear Inteira (PLI) [6] [7] formula e resolve diversos problemas que estão contidos na categoria de Otimização Linear, nos quais algumas ou todas variáveis são restritas a valores inteiros. PLI pode aparecer nomeada também somente como Programação Inteira ou Otimização Inteira ou Otimização Discreta.

Um problema de PLI pode ser escrito em forma matricial como segue-se:

$$\begin{aligned} \max \quad & c^T x + d^T y \\ \text{s.t.} \quad & Ax + Dy \leq b \\ & x \in \mathbb{R}_+^n, y \in \mathbb{Z}_+^p \end{aligned}$$

na qual  $c^T$  é uma matriz  $1 \times n$ ,  $d^T$  é uma matriz  $1 \times p$ ,  $A$  é uma matriz  $m \times n$ ,  $b$  é uma matriz  $m \times 1$ . Estes representando os dados de entrada do problema. Além disso,  $x$  e  $y$  são matrizes  $n \times 1$  e  $p \times 1$ , respectivamente, de variáveis. Tem-se que  $c^T x + d^T y$  é a função objetivo, a qual pode ser de minimização ou maximização (vale ressaltar que maximizar uma função equivale a minimizar o negativo desta função e vice-versa) e  $Ax + Dy \leq b$  são as restrições que as soluções estão sujeitas.

O problema acima descrito é caracterizado como um problema de Programação Linear Inteira Mista (PLIM), pois nem todas as variáveis são inteiras. Quando todas as variáveis estão sujeitas à condição de integralidade, estamos diante de um problema de Programação Linear Inteira Pura (PLIP). Já quando as variáveis assumem, exclusivamente, valores binários, ou seja, 0 (zero) ou 1 (um), temos problema de Programação Inteira Binária ou Programação Inteira 0-1 [8].

Problemas de PLI são geralmente muito mais difíceis de serem resolvidos quando comparados aos problemas de Programação Linear (PL) ordinários. A fim de resolver essa situação, surgiram diversos tipos de algoritmos de PLI que são baseados na exploração do indiscutível sucesso computacional da PL. Segundo [8] baseiam-se em três etapas:

1. Relaxação da região de soluções da PLI, eliminando a restrição de integralidade imposta a todas as variáveis inteiras. Resulta-se em uma PL.

2. Resolução da PL, obtendo a solução ótima contínua.
3. Adição de restrições especiais, com início no ponto ótimo contínuo, que modifiquem iterativamente a região de soluções da PL de maneira que, a certa altura, resultará em um ponto extremo ótimo que satisfaz os requisitos inteiros.

Ignorando-se a etapa 3 para truncar ou arredondar seus valores da solução ótima contínua, também é uma forma de resolução. Porém, não é um procedimento robusto, pois pode gerar solução que não é ótima (destaca-se aqui que não existe maneira fácil de verificar se uma solução viável é ótima) ou, até mesmo, inviável.

Foram desenvolvidos dois métodos para gerar as restrições na etapa 3: são eles o método de planos de corte e o método *branch-and-bound* (B&B). A combinação destes gerou na década de 80 o método *branch-and-cut* [6].

O método de Planos de Corte (Cutting Planes, em inglês) foi o primeiro método a ser desenvolvido e deve-se a [9]. Consiste em introduzir sucessivamente novas restrições na relaxação linear do PLI. Restrições essas que cortam o conjunto das soluções possíveis, eliminando algumas delas e a própria solução ótima do PL, sem contudo eliminar qualquer solução inteira.

O primeiro algoritmo B&B foi desenvolvido por [10], no qual é baseado no conceito de divisão e conquista. Consiste na partição (ramificação) sucessiva do conjunto de soluções possíveis do problema de PLI em subproblemas e na limitação (avaliação) do valor ótimo da função objetivo (limite inferior ou superior se for de maximização ou minimização, respectivamente), de modo a excluir os subproblemas que não contenham a solução ótima. O Algoritmo 1 mostra em pseudocódigo como se comporta o B&B, considerando um problema de maximização.

## 2.2 Combinatória Poliédrica

A Combinatória Poliédrica é baseada na teoria de poliedros e de programação linear, estudando as propriedades de poliedros relacionados ao Problema de Otimização Combinatória (POC) [11]. Vale ressaltar que vários POCs podem ser formulados como um PLI [6]. O presente trabalho foca no POC linear e, deste modo, simplificando a formulação como um PLI.

POC pode ser definido [6] como: sejam um conjunto finito  $N = \{1, \dots, n\}$ , função custo  $c_j : N \rightarrow \mathbb{R}$  para cada  $j \in N$  e um conjunto de soluções viáveis  $\mathcal{F}$ , encontre um  $S \in \mathcal{F}$  e  $S \subseteq N$  que minimize ou maximize  $\sum_{j \in S} c_j$ .

O domínio do POC é finito, podendo assim, na maioria dos casos, gerar ou testar os elementos quanto à sua pertinência a este domínio. Porém, mesmo sendo finito, o

---

**Algoritmo 1: ALGORITMO BRANCH-AND-BOUND**


---

```

1  $z^* \leftarrow -\infty$  (melhor solução até o momento)
2  $x^* \leftarrow \emptyset$  (incumbente)
3 while existir subproblema para ser investigado do
4   Resolva um subproblema pelo método simplex
5   if o subproblema for infactível then
6     | descarte-o
7   else if Se a solução for inteira then
8     | if  $z \text{ de } x^* \leq z^*$  then
9       | descarte o subproblema
10    | else
11      |  $z^* \leftarrow z$ 
12      |  $x^*$  é a nova incumbente
13    | end
14    | Os demais subproblemas com  $z \leq z^*$  são descartados
15  else if Se a solução não for inteira then
16    | if  $z \text{ de } x^* \leq z^*$  then
17      | descarte o subproblema, não há contribuição
18    | else
19      | Escolha um  $x_i$  qualquer que tenha um valor não inteiro na solução
20      | Crie um subproblema adicionando a restrição  $x_i \leq \lfloor x_i^* \rfloor$ 
21      | Crie outro subproblema adicionando a restrição  $x_i \geq \lceil x_i^* \rceil$ 
22    | end
23  end
24 end
25 return  $x^*$  (a incumbente, pois tem o melhor valor)

```

---

tamanho pode ser relativamente grande e, assim, precisando desenvolver um algoritmo que restrinja a busca em um subconjunto deste domínio, de tamanho menor [11].

Associando um POC a um poliedro, podemos fazer uso dos métodos poliédricos, formalizando-os a seguir. Dada a tripla  $(N, c, S)$ , seja  $\mathbb{R}^N$  o espaço vetorial real  $|N|$ -dimensional, no qual cada vetor  $x \in \mathbb{R}^N$  tem seus componentes indexados pelos elementos de  $N$ , i.e.,  $x = (x_j)_{j \in N}$ . Para cada elemento  $S \in \mathcal{F}$ , associamos um ponto em  $\mathbb{R}^N$ , correspondendo ao seu vetor de incidência  $\chi^S = (\chi_j^S)_{j \in N}$ , definido [11] como

$$\chi_j^S := \begin{cases} 1, & \text{se } j \in S \\ 0, & \text{caso contrário} \end{cases}$$

Seja agora o politopo  $P \subseteq \mathbb{R}^N$ , definido como o fecho convexo dos vetores de incidência dos subconjuntos em  $\mathcal{F}$ , i.e.,  $P := \text{conv}\{\chi^S \in \mathbb{R}^N \mid S \in \mathcal{F}\}$ . O conjunto de vértices de  $P$  é precisamente o conjunto dos vetores de incidência dos elementos de  $S \in \mathcal{F}$ . Desta forma, podemos definir o problema original, de forma equivalente, como

$$\min\{c^T x \mid x \in P\}$$

e portanto o objetivo da Combinatória Polidrica é encontrar o sistema de inequações lineares (definem facetas) que determinam a envoltória convexa de um POC [11].

Para melhor entendimento dessas definições e terminologias, e das posteriores que virão no presente trabalho, faz-se necessário outros conceitos prévios de álgebra linear e da teoria de poliedros que serão abordados agora [6] [7] [11] [12].

**Definição 2.1.** Um vetor  $x \in \mathbb{R}^n$  é uma combinação linear dos vetores  $x_1, \dots, x_t \in \mathbb{R}^n$ , se, para algum  $\alpha = (\alpha_1, \dots, \alpha_t) \in \mathbb{R}^t$ ,  $x = \sum_{i=1}^t \alpha_i x_i$ . Tal combinação linear é chamada de afim, se  $\alpha_1 + \dots + \alpha_t = 1$ ; cônica, se  $\alpha_1, \dots, \alpha_t \geq 0$ ; convexa, se for afim e cônica.

**Definição 2.2.** Um conjunto de vetores  $x_1, \dots, x_t \in \mathbb{R}^n$  é Linearmente Independente (LI) se a única solução de  $\sum_{i=1}^t \alpha_i x_i = 0$ , onde  $\alpha_i \in \mathbb{R}$ , for  $\alpha_i = \dots = \alpha_t = 0$ .

**Definição 2.3.** Um conjunto de vetores  $x_1, \dots, x_t \in \mathbb{R}^n$  é Afim Independente (AI) se a solução de  $\sum_{i=1}^t \alpha_i x_i = 0$  e  $\sum_{i=1}^t \alpha_i = 0$ , onde  $\alpha_i \in \mathbb{R}$ , for  $\alpha_i = \dots = \alpha_t = 0$ .

As definições (2.1), (2.2) e (2.3) são para caracterizar a independência afim e dimensão de um poliedro, como será mostrado na definição (2.6).

**Definição 2.4.** Um subconjunto  $P \subseteq \mathbb{R}^n$  é chamado de poliedro se para alguma matriz  $A \in \mathbb{R}^{m \times n}$  e algum vetor  $b \in \mathbb{R}^m$ ,  $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ . Um poliedro definido desta forma pode ser denotado por  $P(A, b)$ .

**Definição 2.5.** Se existe um  $\alpha \in \mathbb{R}$  tal que todo ponto  $x \in P$  satisfaz  $\|x\| \leq \alpha$ , ou seja, um poliedro limitado, então será chamado de politopo.

**Definição 2.6.** Um poliedro  $P$  tem dimensão  $k$ , denotado por  $\dim(P) = k$ , se o número máximo de vetores afim independente em  $P$  é  $k+1$ .

**Definição 2.7.** Uma inequação  $\pi x \leq \pi_0$ , denotada por  $(\pi, \pi_0)$ , é uma inequação válida para  $P \subseteq \mathbb{R}^n$  se  $\pi x \leq \pi_0$  é satisfeita  $\forall x \in P$ .

**Definição 2.8.** Um conjunto  $F \subseteq P$  define uma face de  $P$  se  $F = \{x \in P : \pi x = \pi_0\}$ , para alguma inequação válida  $\pi x \leq \pi_0$ .

**Definição 2.9.** Se  $F$  é uma face de  $P$  e  $\dim(F) = \dim(P) - 1$ , então dizemos que  $F$  é uma faceta de  $P$ .

Já as definições (2.4) e (2.5) apresentam o que são poliedro e politopo, enquanto a (2.7) descreve como identifica-se uma inequação válida de  $P$ , para, a partir disso, em (2.8) indicar o que é uma face de poliedro e, conseqüentemente em (2.9), uma faceta.

**Teorema 2.1.** *Se  $F \neq \emptyset$  é uma face do poliedro  $P(A, b) \subseteq \mathbb{R}^n$ , então  $\dim(F) = n - \text{posto}(A_{(F)})$ , no qual o  $\text{posto}(A_{(F)})$  é o número máximo de colunas LI de  $A$ .*

**Corolário 2.2.** *Seja  $P = P(A, b) \subseteq \mathbb{R}^n$  um poliedro não-vazio, então  $\dim(P) = n - \text{posto}(A_{(P)})$ . E se  $\text{posto}(A_{(P)}) = 0$ , então  $P$  tem dimensão plena, i.e,  $\dim(P) = n$*

**Proposição 2.3.** *Se  $F$  é uma faceta de  $P$ , existe uma desigualdade  $(\pi, \pi_0)$  que a representa.*

**Proposição 2.4.** *Para cada faceta  $F$  de  $P$ , uma das desigualdades que representa  $F$  é necessária na descrição de  $P$ .*

**Proposição 2.5.** *Toda desigualdade que representa uma face de  $P$  com dimensão menor que  $\dim(P) - 1$  é irrelevante na descrição de  $P$ .*

O teorema (2.1) e o corolário (2.2) facilitam a calcular a dimensão do poliedro, e as proposições (2.3), (2.4) e (2.5) explicam que as facetas são as únicas desigualdades necessárias para descrever o poliedro  $P$ , ou seja, a melhor descrição possível.

## 2.3 Trabalhos Relacionados

Este Subcapítulo foi dividido em duas partes, no qual a primeira parte mostrará trabalhos relacionados à abordagem poliédrica para o tratamento de POCs, enquanto a segunda parte apresentará trabalhos relacionados ao RTVP.

### 2.3.1 Estudo Poliedral

Rado [13] foi o primeiro a utilizar explicitamente métodos de combinatória poliédrica, fazendo uso do Lema de Farkas indiretamente. Porém, somente depois que deu-se concretamente o desenvolvimento da Combinatória Poliédrica, podendo ser ressaltado três marcos.

O primeiro marco aconteceu nos anos 50 com o descobrimento do método simplex. A observação de que muitos POCs podem ser modelados como PLIs, também que suas respectivas matrizes de restrições admitem unimodularidade total, e assim serem tratados como PL e resolvidos pelo método simplex e dualidade, resultou norteando diversas pesquisas [14] [15].

Já o segundo marco foi nos anos 60 com a obtenção de uma descrição linear completa de tamanho exponencial para o politopo dos emparelhamentos de um grafo [16]. Observou-se que uma descrição de tamanho exponencial poderia ser útil para desenvolvimento de algoritmos polinomiais.

O terceiro marco surgiu em 1979 com a apresentação do método elipsoide para PL [17]. As contribuições foram que PLs podem ser resolvidos em tempo polinomial e a prova da equivalência de resolução para um problema de otimização e de separação, ou seja, é possível resolver eficientemente um PL sabendo resolver eficientemente o problema de separação.

Tal prova de equivalência incrementou o desenvolvimento de algoritmos na área de Combinação Poliédrica, pois facilitou conseguir algoritmos polinomiais para certos problemas devido à capacidade de associar poliedros a POCs e resolver o problema de separação coincidente. Ajudou também a tratar problemas NP-difíceis de grande porte, nos quais os métodos anteriores eram inviáveis [18] [19] [20].

Os problemas NP-difíceis têm semelhança no fato de que conhece-se apenas uma descrição parcial dos politopos associados a esses problemas, contudo, para maioria deles, o problema de separação foi resolvido apenas para classes especiais de faceta.

Dantzig [21] foi o primeiro a utilizar uma abordagem poliédrica para um problema NP-difícil, o problema do caixeiro viajante. E [22] aprofundou-se mais no estudo da estrutura facial dos politopos associados ao problema do caixeiro viajante. Para fim de destaque, outros problemas NP-difíceis também foram atacados, como por exemplo, o problema das mochilas múltiplas [23] e problema livre de garra [24].

### **2.3.2 Problema da Variabilidade do Tempo de Resposta**

Corominas et al. [3] foi o primeiro a apresentar o RTVP, no qual definiu formalmente o problema e sua complexidade computacional, demonstrando que o problema é NP-difícil. O modelo matemático baseado em PLIM, no entanto, só foi capaz de lidar com pequenas instâncias do problema. Além disso, também propôs algumas heurísticas para o problema, baseadas em um procedimento de troca simples na solução inicial.

Corominas et al. [25] aprimorou o modelo exato e, dessa forma, instâncias maiores puderam ser resolvidas até a otimalidade. Mostrou a importância da modelagem e o impacto que a reformulação, acréscimo de restrições redundantes e a eliminação de simetrias têm na efetividade de um PLIM.

Em razão do RTVP se tratar de um POC NP-difícil, intensificaram-se na literatura estudos e desenvolvimentos de algoritmos fundamentados em soluções heurísticas e metaheurísticas. Dentre eles, destacou-se [26] que propôs o algoritmo Procedimento de

Busca Adaptativa Gulosa e Randômica (GRASP, do inglês *greedy randomized adaptive search procedure*) e multi-start para resolver o RTVP. Outro foi [27] que apresentou uma forma de resolver o RTVP por meio do algoritmo do Mecanismo Eletromagnético (do inglês, *electromagnetism-like mechanism*) para comparar com os algoritmos já implementados até então. O [28] mostrou um algoritmo de Busca Tabu para o RTVP. Além destes, temos o [29] que adaptou um Algoritmo Genético para resolver o RTVP.

García-Villoria & Pastor [30] propuseram uma versão do RTVP com enfoque na métrica do erro de *throughput* do escalonamento *stride*, denominado de minimização do tempo de resposta máximo. Foram apresentadas uma nova formulação matemática e soluções baseadas em metaheurísticas.

### 3 METODOLOGIA

Descrevem-se neste capítulo a tipologia de pesquisa, o politopo para o problema, as equações lineares e inequações lineares do politopo, e a proposta de modelo.

#### 3.1 Tipologia de Pesquisa

O presente estudo objetivou compreender o poliedro do RTVP relacionando-o ao politopo de sequenciamento de símbolos, e caracteriza-se, quanto à natureza, como aplicada, visto que buscou solucionar e gerar conhecimento sobre um problema já consolidado na literatura que tem diversas aplicações.

Quanto aos objetivos, classifica-se como descritiva, uma vez que objetiva descrever as características do problema da variabilidade do tempo de resposta como um poliedro, relacionando-o com um problema de otimização combinatória.

Quanto à abordagem, define-se como quantitativa, pois a análise dos conceitos do problema da variabilidade do tempo de resposta e da combinatória poliédrica realizou-se por meio de ferramentas matemáticas e/ou técnicas computacionais, apresentando-a de forma objetiva.

Quanto aos procedimentos, determina-se como bibliográfica, porque baseia-se em informações e conhecimentos prévios sobre o problema, procuradas em referências teóricas já analisadas, com o fim de formular o RTVP como um poliedro juntamente com suas hipóteses.

#### 3.2 Politopo de Sequenciamento de Símbolos

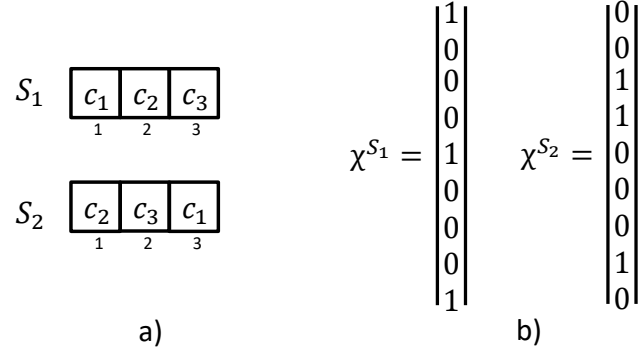
Seja  $\mathcal{C} = \{1, 2, \dots, d\}$  o conjunto de cópias de símbolos e  $H = \{1, 2, \dots, d\}$  o conjunto de slots que as cópias de símbolos serão atribuídas. Seja  $P_d$  a envoltória convexa dos vetores de incidência  $\chi^S$ , na qual  $S$  representa uma sequência de atribuições de cada cópia de símbolo  $c \in \mathcal{C}$  a um único slot  $h \in H$  e cada slot  $h$  recebe uma única cópia  $c$ , sendo  $\chi_{c,h}^S = 1$  a atribuição da cópia  $c \in \mathcal{C}$  no slot  $h \in H$  e  $\chi_{c,h}^S = 0$  em caso contrário. Sendo assim, temos:

$$P_d = \text{conv}\{\chi^S \in \{0, 1\}^{d^2} \mid S \text{ é uma sequência de cópias } c \in \mathcal{C} \text{ atribuídas a slots } h \in H\},$$

logo,  $P_d$  é definido como o *politopo de sequenciamento de símbolos* (PSS). O PSS contém todas as possíveis soluções do RTVP e, deste modo, podem ser relacionados.



A Figura 1 ilustra duas seqüências de cópias de símbolos  $S_1 = \{(1, 1), (2, 2), (3, 3)\}$  e  $S_2 = \{(2, 1), (3, 2), (1, 3)\}$  e seus vetores de incidência.



**Figura 1:** (a) Sequências de símbolos  $S_1$  e  $S_2$  e (b) vetores de incidência  $\chi^{S_1}$  e  $\chi^{S_2}$ .

O PSS pode ser descrito como um problema de otimização que procura o vetor de incidência  $x^T = (x_1, x_2, \dots, x_{d^2})$  com menor custo  $w x^T$  associado:

$$\begin{aligned} \min & \quad w x^T \\ \text{s.t.} & \quad x \in P_d \end{aligned}$$

A fim de aplicar técnicas de PLI para o PSS, é necessário descrever  $P_d$ , que é um politopo  $d^2$ -dimensional, em equações e inequações lineares.

### 3.3 Equações lineares de $P_d$

**Teorema 3.1.** *As seguintes equações lineares são válidas por todo vetor de  $P_d$ :*

$$\sum_{\forall h \in H} x_{d,h} = 1, \quad (1)$$

$$\sum_{\forall h \in H} x_{c,h} = \sum_{\forall h \in H} x_{c+1,h}, \forall c \in \mathcal{C} \setminus \{d\} \quad (2)$$

*Demonstração.* As equações (1) e (2) são complementares. A equação (1) faz com que o último símbolo esteja atribuído a um e somente um slot e, a partir desta atribuição, a equação (2) faz com que cada símbolo do penúltimo ao primeiro também esteja atribuído a um e somente um slot. Nota-se que pela definição do problema, isto já está estabelecido,

tornando assim as equações (1) e (2) válidas para  $P_d$ , ou seja, não cortam nenhuma solução viável do problema.

□

Seja  $\mathcal{P} = \{(S_1, T_1), (S_2, T_2), \dots, (S_{d-1}, T_{d-1})\}$  o conjunto de  $d - 1$  particionamentos de  $H$ . Sendo  $\mathcal{P}$  construído pelo Algoritmo 2. E seja  $I = \mathcal{C} \setminus \{d - 1, d\}$  um subconjunto de  $\mathcal{C}$ .

---

**Algoritmo 2:** ALGORITMO PARA CONSTRUÇÃO DO CONJUNTO  $\mathcal{P}$  DE PARTIÇÕES DE  $H$

---

```

1  $\mathcal{P} = \emptyset, S_0 = H, T_0 = \emptyset$ 
2 for  $h \in H \setminus \{d\}$  do
3    $S_h = S_{h-1} \setminus \{h\}$ 
4    $T_h = T_{h-1} \cup \{h\}$ 
5    $\mathcal{P} = \mathcal{P} \cup (S_h, T_h)$ 
6 end
7 return  $\mathcal{P}$ 

```

---

**Teorema 3.2.** *Seja  $(S_i, T_i) \in \mathcal{P}$ , a correspondente equação*

$$\sum_{c \in I} \sum_{h \in S_i} x_{c,h} - (|I| - |T_i|) \sum_{h \in S_i} x_{d-1,h} = (|S_i| - 1) \sum_{h \in T_i} x_{d-1,h} + \sum_{h \in T_i} x_{d,h}, \quad (3)$$

*é válida para  $P_d$ .*

*Demonstração.* A prova de validade da equação 3 está dividida em 4 casos.

*Caso 1:* Se os slots  $d - 1$  e  $d \in T_i$ , então  $d - 1 \notin S_i$ , logo,  $\sum_{h \in T_i} x_{d-1,h} = \sum_{h \in T_i} x_{d,h} = 1$  e  $\sum_{h \in S_i} x_{d-1,h} = 0$ . Sendo assim, a equação (3) torna-se:

$$\sum_{c \in I} \sum_{h \in S_i} x_{c,h} = |S_i|, \quad (3.1)$$

logo, a equação (3.1) é válida se  $|S_i| \leq |I|$ . Sendo  $|I| = d - 2$  e  $|S_i| = d - |T_i|$ . Como os slots  $d - 1$  e  $d \in T_i$ , então  $|T_i| \geq 2$ , logo,  $|S_i| \leq d - 2$  e, portanto,  $|S_i| \leq |I|$ . Como as cópias  $d - 1$  e  $d \notin I$ , a equação (3.1) obriga que  $|S_i|$  cópias de  $I$  estejam atribuídos na partição de slots  $S_i$ . Fato obrigatório por definição para toda solução de  $P_d$ .

*Caso 2:* Se os slots  $d - 1$  e  $d \notin T_i$ , então os slots  $d - 1$  e  $d \in S_i$ , logo,  $\sum_{h \in T_i} x_{d-1,h} = \sum_{h \in T_i} x_{d,h} = 0$  e  $\sum_{h \in S_i} x_{d-1,h} = 1$ . Sendo assim a equação (3) torna-se:

$$\sum_{c \in I} \sum_{h \in S_i} x_{c,h} = (|I| - |T_i|), \quad (3.2)$$

logo, a equação (3.2) é válida quando  $|S_i| \geq |I| - |T_i|$ . Como  $|I| = d - 2$  e  $|S_i| + |T_i| = d$ , então  $|S_i| + |T_i| = |I| + 2$ . Assim sendo,  $|S_i| = |I| - |T_i| + 2$  e, portanto,  $|S_i| \geq |I| - |T_i|$ . Como as cópias  $d - 1$  e  $d \notin I$ , a equação (3.2) obriga que  $|I| - |T_i|$  cópias de  $I$  sejam atribuídas na partição de slots  $S_i$ , pois a partição  $T_i$  receberá  $|T_i|$  cópias de  $I$ . Fato obrigatório por definição para toda solução de  $P_d$ .

Caso 3: Se os slots  $d - 1 \in T_i$  e  $d \notin T_i$ , então  $d - 1 \notin S_i$  e  $d \in S_i$ , logo,  $\sum_{h \in T_i} x_{d-1,h} = 1$  e  $\sum_{h \in S_i} x_{d-1,h} = \sum_{h \in T_i} x_{d,h} = 0$ . Sendo assim, a equação (3) torna-se:

$$\sum_{c \in I} \sum_{h \in S_i} x_{c,h} = |S_i| - 1, \quad (3.3)$$

logo, temos que (3.3) é válida se  $|S_i| - 1 \leq |I|$ . Como  $|S_i| + |T_i| = d$  e sabendo que  $d - 1 \in T_i$ , então  $|T_i| \geq 1$ . Logo,  $|S_i| \leq d - 1$ . Como  $|I| = d - 2$ , então  $|S_i| - 1 \leq |I|$ . Como o slot  $d \in S_i$  e a cópia  $d \notin I$ , então a equação (3.3) obriga que  $|S_i| - 1$  elementos de  $I$  estejam na partição de slots  $S_i$ . Fato obrigatório por definição para toda solução de  $P_d$ .

Caso 4: Se os slots  $d - 1 \notin T_i$  e  $d \in T_i$ , então  $d - 1 \in S_i$  e  $d \notin S_i$ , logo,  $\sum_{h \in S_i} x_{d-1,h} = \sum_{h \in T_i} x_{d,h} = 1$  e  $\sum_{h \in T_i} x_{d-1,h} = 0$ . Sendo assim, a equação (3) torna-se:

$$\sum_{c \in I} \sum_{h \in S_i} x_{c,h} = (|I| - |T_i|) + 1, \quad (3.4)$$

logo, temos que (3.4) é válido para  $|S_i| \geq |I| - |T_i| + 1$  e  $|I| \geq |I| - |T_i| + 1$ . Como  $|T_i| \geq 1$ , pois o slot  $d \in T_i$ , então é verdade que  $|I| \geq |I| - |T_i| + 1$ . Do caso 2 temos que  $|S_i| = |I| - |T_i| + 2$ , portanto garantimos que  $|S_i| \geq |I| - |T_i| + 1$ . Como o slot  $d - 1 \notin T_i$ , a equação (3.4) obriga que  $(|I| - |T_i| + 1)$  cópias de  $I$  estejam atribuídas em  $S_i$ , pois  $T_i$  receberá  $|T_i|$  cópias de  $I$  e já tem a cópia  $d$  atribuída, obrigando assim que uma cópia a mais seja atribuída em  $S_i$ . Fato obrigatório por definição para toda solução de  $P_d$ .

Como para todos os casos possíveis a equação ainda é válida, logo a equação (3) é válida para  $P_d$ .

□

### 3.4 Inequações lineares de $P_d$

**Teorema 3.3.** *Para todo  $h \in H \setminus \{1\}$ , a correspondente inequação*

$$\sum_{c \in \mathcal{C} \setminus \{1\}} x_{c,h} \leq 1, \quad (4)$$

*é válida para  $P_d$ .*

*Demonstração.* A inequação (4) diz que para cada slot, menos o primeiro, terá de ter no máximo um símbolo atribuído. Isto já é prescrito na própria definição do problema, na qual diz que para cada slot atribui um e somente um símbolo. Logo, a inequação (4) é válida para  $P_d$ . □

**Teorema 3.4.** *Para todo  $c \in \mathcal{C} \setminus \{1\}$ , a correspondente inequação*

$$\sum_{h \in H \setminus \{1\}} x_{c,h} \leq 1, \quad (5)$$

*é válida para  $P_d$ .*

*Demonstração.* A inequação (5) diz que cada símbolo, menos o primeiro, terá de estar atribuído em no máximo um slot. Novamente já está prescrito na definição do problema, na qual diz que um símbolo deve estar atribuído a um e somente um slot. Logo, a inequação (5) é válida para  $P_d$ . □

**Teorema 3.5.** *A inequação*

$$\sum_{c \in \mathcal{C} \setminus \{1\}} \sum_{h \in H \setminus \{1\}} x_{c,h} \geq d - 2, \quad (6)$$

*é válida para  $P_d$ .*

*Demonstração.* A inequação (6) garante que nos últimos  $d - 1$  slots, terá de ter pelo menos  $d - 2$  atribuições de símbolos, e também que dos últimos  $d - 1$  símbolos, pelo menos  $d - 2$  terão de estar atribuídos. Nota-se que pela definição do problema, em  $d - 1$  slots tem de haver exatamente  $d - 1$  símbolos. Portanto a inequação (6) não invalida nenhuma solução pois  $d - 2 < d - 1$ , logo é válida para  $P_d$ . □

**Conjectura 3.6.** *As inequações (4), (5) e (6) definem facetas de  $P_d$ .*

### 3.5 Proposta de Modelo

O processo de linearização da função objetivo do RTVP foi proposto por [3], no qual ele introduziu a variável  $\delta_{ik}^j$  ( $i = 1, \dots, n; k = 1, \dots, m_i; j = 1, \dots, UB_i$ ) que assume valor 1 se e somente se a distância entre as cópias  $k$  e  $k + 1$  do símbolo  $i$  é igual a  $j$ , caso contrário assume valor zero, sendo  $UB_i = M - m_i + 1$  ( $i = 1, \dots, n$ ). Também incluiu mais 3 (três) restrições e, assim, o modelo proposto neste presente trabalho pode ser escrito como:

$$\begin{aligned} \min RTV = & \sum_{(i,k),j} j^2 \cdot \delta_{ik}^j - \sum_i m_i \cdot \bar{d}_i^2 \\ \text{s.a.} \end{aligned}$$

$$(1), (2), (3), (4), (5), (6)$$

$$\sum_{h \in H} h \cdot x_{(i,k+1),h} - \sum_{h \in H} h \cdot x_{(i,k),h} = \sum_{j=1}^{UB_i} j \cdot \delta_{ik}^j \quad \forall i \in \{1, \dots, n\}; \forall k \in \{1, \dots, m_i - 1\} \quad (7)$$

$$M - \sum_{h \in H} h \cdot x_{(i,m_i),h} + \sum_{h \in H} h \cdot x_{(i,1),h} = \sum_{j=1}^{UB_i} j \cdot \delta_{im_i}^j \quad \forall i \in \{1, \dots, n\}; \forall k \in \{1, \dots, m_i - 1\} \quad (8)$$

$$\sum_{j=1}^{UB_i} \delta_{ik}^j = 1 \quad \forall i \in \{1, \dots, n\}; \forall k \in \{1, \dots, m_i\} \quad (9)$$

$$x, \delta \in \{0, 1\}$$

As restrições (1), (2), (3), (4), (5) e (6) são obtidas da descrição do PSS e as (7), (8) e (9) foram as propostas por [3] para a linearização da função objetivo. Vale salientar que cada par  $(i, k)$  representa um símbolo  $c \in \mathcal{C}$ .

As restrições (7) e (8) garantem que a distância entre as cópias  $k$  e  $k + 1$  do símbolo  $i$  seja igual a um valor inteiro  $j \in [1, UB_i]$ , sendo que a restrição (8) se refere à distância entre a primeira cópia do símbolo  $i$  em um ciclo e a última cópia do mesmo símbolo no ciclo anterior. Já a restrição (9) garante que a distância entre as cópias  $k$  e  $k + 1$  do símbolo  $i$  seja obtida com um e apenas um valor inteiro.

## 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Para utilizar a modelagem matemática obtida pela descrição do poliedro do RTVP realizada na metodologia, faz-se uso da relação entre POC linear e PLI, e, por conseguinte, a associação de um POC a um poliedro, apresentadas na fundamentação teórica.

Com a finalidade de avaliar essa modelagem, foi utilizado o conjunto de instâncias para o método exato geradas aleatoriamente por [3] e comparando-a com o método exato dele.

No total há 320 instâncias agrupadas pelo número total de cópias a ser sequenciado ( $d$ ) e, mais ainda, agrupadas pelo número de símbolos ( $n$ ). Tendo  $d \in (20, 60)$ , e  $n \in [3, 12]$ , totalizou 16 grupos com, em média, 20 instâncias cada.

Todos os métodos estudados neste trabalho foram desenvolvidos em linguagem C++, utilizando o compilador g++ versão 5.4.0 (opção de compilação -O3), com o auxílio do resolvidor matemático CPLEX versão 12.7. Os experimentos computacionais foram executados em um computador com processador Intel Core I7 de 64 bits, com CPU contendo 8 cores de 3.4 GHz, 16 GB de RAM e o sistema operacional Ubuntu 16.04. Os experimentos foram realizados com tempo limite de duas horas para cada instância.

A Tabela 1 apresenta os resultados dos experimentos computacionais. Cada linha representa um grupo de instâncias, como, por exemplo, o grupo d20\_d30\_n3\_n7 que descreve o grupo com  $d \in (20, 30)$  e  $n \in [3, 7]$ . Os dois conjuntos de colunas principais representam o modelo apresentado por [3] e o modelo apresentado nesse trabalho, respectivamente. Para cada conjunto de coluna (Corominas e PSS) foi calculada a média do *lower bound* (LB raiz) e tempo de execução em segundos na raiz da árvore (T raiz(s)), *lower bound* (LB), *upper bound* (UB), GAP (calculado como  $\frac{UB-LB}{LB} * 100$ ), número de nós (#Nós) e o tempo de execução total em segundos (T (s)) para cada grupo de instância.

Na Tabela 1 é possível ver que o PSS fornece um *lower bound* inicial maior (i.e., mais perto do ótimo) em quase todos os grupos de instâncias, dos 16 perdendo apenas em 2. Em geral, leva mais tempo para montar a raiz, mas em compensação converge mais rápido no tempo total, como pode ser observado no grupo de instância d45\_d50\_n8\_n12, no qual o modelo da literatura levou 20,441 segundos para montar a raiz, enquanto o PSS levou 28,597 segundos, mas para o tempo total foi gasto 44484,37 e 28150,064 segundos, respectivamente (destacados na tabela com sublinha).

O *lower bound* ser maior e o *upper bound* ser menor (reduzindo o intervalo do valor ótimo) para as instâncias mais difíceis, juntamente com menor número de nós pode explicar a convergência mais rápida do PSS, pois reduz o número de execuções Simplex do algoritmo *branch-and-bound* para cada construção de um novo nó na busca de garantir a integralidade das soluções. Isso é verificado nos resultados, uma vez que o PSS tem o

maior número de nós em apenas um grupo de instância.

Outra informação que reforça a convergência é o GAP, o qual indica a porcentagem do *lower bound* que resta para igualar-se ao *upper bound*. Além de ganharmos em 11 grupos de instâncias e empatarmos em 3, o modelo da literatura apresenta uma mediana de 36,505%, enquanto o PSS é apenas 23,425%. Sendo assim, os resultados corroboram na conjectura de que as inequações são facetas do politopo.

**Tabela 1: Resultados Computacionais**

Instâncias	Corominas							PSS						
	LB raiz	T raiz (s)	LB	UB	GAP (%)	#Nós	T (s)	LB raiz	T raiz (s)	LB	UB	GAP (%)	#Nós	T (s)
d20_d30_n3_n7	<b>-11,323</b>	<b>2,735</b>	<b>-7,142</b>	<b>7,995</b>	<b>14,293</b>	19.218,727	77,411	-13,968	3,023	-10,233	<b>7,995</b>	19,438	<b>6.629,682</b>	<b>24,240</b>
d20_d30_n8_n12	2,967	<b>1,636</b>	<b>4,078</b>	<b>4,078</b>	<b>0</b>	1.039,000	5,988	<b>3,382</b>	2,324	<b>4,078</b>	<b>4,078</b>	<b>0</b>	490,111	<b>3,301</b>
d30_d35_n3_n7	<b>5,412</b>	<b>7,516</b>	<b>11,517</b>	<b>11,517</b>	<b>0</b>	164.230,158	1.051,145	-7,746	8,907	-1,851	<b>11,517</b>	11,215	<b>44.135,947</b>	<b>259,821</b>
d30_d35_n8_n12	4,805	<b>6,273</b>	<b>7,186</b>	<b>7,186</b>	<b>0</b>	70.313,810	715,226	<b>4,940</b>	8,787	<b>7,186</b>	<b>7,186</b>	<b>0</b>	<b>11.987,381</b>	<b>89,390</b>
d35_d40_n3_n7	5,904	<b>9,540</b>	13,702	<b>15,424</b>	15,012	1.799.398,880	17.340,854	<b>5,971</b>	14,189	<b>14,989</b>	<b>15,424</b>	<b>2,97</b>	<b>823.641,800</b>	<b>8.600,854</b>
d35_d40_n8_n12	4,877	<b>10,340</b>	<b>10,210</b>	<b>10,210</b>	<b>0</b>	546.988,400	4.384,168	<b>4,927</b>	14,070	<b>10,210</b>	<b>10,210</b>	<b>0</b>	<b>102.970,600</b>	<b>808,546</b>
d40_d45_n3_n7	<b>7,788</b>	<b>11,366</b>	13,538	<b>16,788</b>	26,165	1.524.418,688	25.896,261	<b>7,788</b>	15,982	<b>15,271</b>	<b>16,788</b>	<b>9,890</b>	<b>951.146,688</b>	<b>14.780,376</b>
d40_d45_n8_n12	5,696	<b>14,620</b>	10,497	<b>12,862</b>	24,649	1.801.524,333	29.359,042	<b>5,697</b>	20,491	<b>11,846</b>	<b>12,862</b>	<b>8,689</b>	<b>868.481,750</b>	<b>15.409,005</b>
d45_d50_n3_n7	<b>8,626</b>	<b>14,751</b>	12,972	<b>18,126</b>	46,844	1.557.618,750	35.994,611	<b>8,626</b>	20,735	<b>14,590</b>	<b>18,126</b>	<b>27,411</b>	<b>1.199.821,400</b>	<b>28.021,078</b>
<u>d45_d50_n8_n12</u>	<b>6,056</b>	<u><b>20,441</b></u>	9,755	<b>14,856</b>	63,635	1.836.870,700	<u>44.484,370</u>	<b>6,056</b>	<u>28,597</u>	<b>12,266</b>	14,956	<b>29,040</b>	<b>1.150.955,150</b>	<u><b>28.150,064</b></u>
d50_d55_n3_n7	<b>8,827</b>	<b>17,650</b>	12,078	<b>19,523</b>	80,067	975.104,435	31.533,099	<b>8,827</b>	27,625	<b>13,051</b>	19,610	<b>56,988</b>	<b>886.991,783</b>	<b>30.219,604</b>
d50_d55_n8_n12	<b>8,314</b>	<b>21,660</b>	10,079	<b>17,726</b>	79,992	1.822.813,588	51.801,529	<b>8,314</b>	36,812	<b>12,035</b>	17,844	<b>53,457</b>	<b>1.513.626,059</b>	<b>47.430,889</b>
d55_d60_n3_n7	<b>9,332</b>	<b>24,448</b>	12,006	21,542	98,821	741.013,053	39.204,328	<b>9,332</b>	34,058	<b>13,234</b>	<b>21,437</b>	<b>73,397</b>	<b>646.432,316</b>	<b>37.169,177</b>
d55_d60_n8_n12	<b>7,994</b>	<b>26,120</b>	9,124	20,756	146,884	1.124.904,619	51.915,490	<b>7,994</b>	40,787	<b>10,589</b>	<b>20,661</b>	<b>109,772</b>	<b>1.016.428,381</b>	<b>51.074,654</b>
d60_d65_n3_n7	<b>10,918</b>	<b>31,136</b>	12,681	<b>24,668</b>	114,963	<b>623.996,833</b>	<b>39.955,744</b>	<b>10,918</b>	35,027	<b>12,972</b>	<b>24,668</b>	<b>105,531</b>	649.402,875	40.034,051
d60_d65_n8_n12	<b>8,925</b>	32,608	9,277	27,300	212,262	1.005.598,562	<b>54.638,705</b>	<b>8,925</b>	<b>17,146</b>	<b>9,830</b>	<b>26,675</b>	<b>183,995</b>	<b>766.113,250</b>	55.769,291

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho trata do Problema da Variabilidade do Tempo de Resposta, o qual consiste em dado um conjunto de cópias de símbolos e um conjunto de slots, atribuir para cada slot uma e somente uma cópia de símbolo e, simultaneamente, atribuir para cada símbolo um e somente um slot, objetivando minimizar a variabilidade do tempo de resposta total. Este tipo de problema possui muitas aplicações reais e é inédito um estudo poliédrico para ele.

Um estudo poliédrico do RTVP foi proposto. Foi visto que esse problema pode ser descrito por meio de equações e inequações lineares, provando-se que estas não anulam nenhuma solução viável do problema, ou seja, são válidas. Também foi apresentado, como conjectura, que as inequações são facetas, para, por fim, poder adaptar o problema em um modelo matemático baseado em PLI e comparar os resultados com um método exato já implementado da literatura.

Nos resultados, dividiu-se 320 instâncias pelo número total de cópias a ser sequenciado e pelo número de símbolos, totalizando em 16 grupos. Observou-se que o modelo PSS, em média, saiu-se bem para a resolução das instâncias: convergiu mais rápido, *lower bound* e *upper bound* mais próximos do valor ótimo, com porcentagem do GAP reduzida, com número de nós reduzido, *lower bound* inicial melhor, evidenciando assim na probabilidade de que as inequações são facetas do politopo.

Como propostas de trabalhos futuros tem-se: (i) provar a dimensão do politopo do RTVP; (ii) provar, em definitivo, que as inequações apresentadas neste trabalho são facetas; (iii) propor um algoritmo baseado em planos de corte; e (iv) encontrar novas facetas para o RTVP.



## REFERÊNCIAS

- [1] PESSOA, Bruno Jefferson de Sousa. **Problema das Sequências Justas Ponderadas**. 2017. Tese (doutorado) – Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica e da Computação, Universidade Federal do Rio Grande do Norte, Natal, 2017.
- [2] PINEDO, Michael L.. **Scheduling: Theory, Algorithms, and Systems (3rd ed.)**. New York: Springer Publishing Company; Incorporated, 2008.
- [3] COROMINAS, Albert, KUBIAK, Wieslaw, e MORENO, Natalia. Response time variability. **J. Scheduling**, v. 10, n. 02, p. 97-110, 2007.
- [4] MONDEN, Yasuhiro. **Toyota production system: practical approach to production management**. Norcross: Industrial Engineering and Management Press; Institute of Industrial Engineers, 1983. Bibliography: p. 233-246.
- [5] HERRMANN, Jeffrey W.. **Generating Cyclic Fair Sequences using Aggregation and Stride Scheduling** (Technical report tr 2007-12). University of Maryland, USA, 2007.
- [6] WOLSEY, Laurence A.. **Integer Programming**. New York: John Wiley & Sons, Inc., 1998.
- [7] SCHRIJVER, Alexander. **A Theory of Linear and Integer Programming**. New York: John Wiley & Sons, Inc., 1986.
- [8] TAHA, Hamdy A.. **Operations Research: An Introduction (8th Edition)**. New Jersey: Pearson Prentice-Hall, 2006.
- [9] GOMORY, R. E.. Outline of an Algorithm for Integer Solutions to Linear Programs. **Bulletin of the American Mathematical Society**, v. 64, n. 05, p. 275-278, m. 09, 1958.
- [10] LAND, A. e DOIG, A.. An Automatic Method of Solving Discrete Programming Problems. **Econometrica**, v. 28, n. 03, p. 497-520, 1958.
- [11] FERREIRA, R. e WAKABAYASHI, Y.. **Combinatória Polédrica e Planos-de-Corte Faciais**. Campinas - SP: X Escola de Computação, 1996.
- [12] NEMHAUSER, G. L. e WOLSEY, L.A.. **Integer and Combinatorial Optimization**. New York: John Wiley & Sons, Inc., 1999.
- [13] RADO, R.. Theorems on Linear Combinatorial Topology and General Measure. **Annals of Mathematics**, v. 44, n. 02, p. 228-270, 1943.

- [14] DANTZIG, G. B.. **Application of the Simplex Method to a Transportation Problem**. In: Koopmans, T.C.. Activity Analysis of Production and Allocation. New York: John Wiley & Sons, Inc., 1951.
- [15] FORD, L. R. e FULKERSON, D. R.. Maximal Flow Through a Network. **Canadian Journal of Mathematics**, v. 08, n. 03, p. 399-404, 1956.
- [16] EDMONDS, Jack. Maximum matching and a polyhedron with 0, 1-vertices. **Journal of Research of the National Bureau of Standards B**, v. 64, n. 01-02, p. 125-130, 1965.
- [17] KHACHIYAN, L. G.. A Polynomial Algorithm in Linear Programming. **Soviet Mathematics Doklady**, v. 20, p. 191-194, 1979.
- [18] GRÖTSCHEL, M., LOVÀSZ, L. e SCHRIJVER, A.. The Ellipsoid Method and its Consequences in Combinatorial Optimization. **Combinatorica**, v. 1, p. 169-197, m. 06, 1981.
- [19] GRÖTSCHEL, M., LOVÀSZ, L. e SCHRIJVER, A.. Corrigendum to our paper "The ellipsoid method and its consequences to combinatorial optimization". **Combinatorica**, v. 4, p. 291-295, m. 12, 1984.
- [20] GRÖTSCHEL, M., LOVÀSZ, L. e SCHRIJVER, A.. **Geometric Algorithms and Combinatorial Optimization**. Berlin: Springer, 1988.
- [21] DANTZIG, G., FULKERSON, R. e JOHNSON, S.. Solution of a Large-Scale Traveling-Salesman Problem. **Journal of the Operations Research Society of America**, v. 2, n. 4, p. 393-410, m. 11, 1954.
- [22] GRÖTSCHEL, M.. **Polyedrische Charakterisierungen kombinatorischer Optimierungprobleme**. Meisenheim am Glan: Hain, 1977.
- [23] FERREIRA, C. E.. **On combinatorial optimization problems arising in computer system design**. 1994. Tese (doutorado) – Technische Universität Berlin, 1994.
- [24] FRAGOSO, F. C., SOUSA FILHO, G. F. e PROTTI, F.. Algoritmos de Planos de Corte para o Problema Livre de Garra. In: LI Simpósio Brasileiro de Pesquisa Operacional, Limeira - SP. **Anais do SBPO....** Sociedade Brasileira de Pesquisa Operacional - SOBRAPO, 2019.
- [25] COROMINAS, Albert, KUBIAK, Wieslaw, e PASTOR, Rafael. Mathematical programming modeling of the Response Time Variability Problem. **European Journal of Operational Research**, v. 200, n. 02, p. 347-357, 2010.

- [26] COROMINAS, Albert, GARCÍA-VILLORIA, Albert e PASTOR, Rafael. **Solving the Response Time Variability Problem by means of Multi-start and GRASP metaheuristics**. In: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence. Artificial Intelligence Research and Development. Sant Martí d'Empúries, 2008.
- [27] GARCÍA-VILLORIA, Albert e PASTOR, Rafael. Solving the Response Time Variability Problem by means of the Electromagnetism-like Mechanism. **International Journal of Production Research**, v. 48, 2008.
- [28] COROMINAS, Albert, GARCÍA-VILLORIA, Albert e PASTOR, Rafael. Using Tabu Search for the Response Time Variability Problem. In: XIII Congreso de Ingeniería de Organización, Barcelona-Terrassa. **Anais...**, 2009.
- [29] GARCÍA-VILLORIA, Albert e PASTOR, Rafael. Solving the response time variability problem by means of a genetic algorithm. **European Journal of Operational Research**, v. 202, n. 02, p. 320-327, 2010.
- [30] GARCÍA-VILLORIA, Albert e PASTOR, Rafael. Minimising maximum response time. **Computers & Operations Research**, v. 40, n. 10, p. 2314-2321, 2013.