

---

Abordagens de solução para o problema de  
alocação de aulas a salas

*Rafael Bernardo Zanetti Cirino*

---



SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 05/07/2016

Assinatura: \_\_\_\_\_

**Rafael Bernardo Zanetti Cirino**

## Abordagens de solução para o problema de alocação de aulas a salas

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Maristela Oliveira dos Santos

**USP – São Carlos**  
**Julho de 2016**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi  
e Seção Técnica de Informática, ICMC/USP,  
com os dados fornecidos pelo(a) autor(a)

C578a      Cirino, Rafael Bernardo Zanetti  
Abordagens de solução para o problema de alocação  
de aulas a salas / Rafael Bernardo Zanetti Cirino;  
orientadora Maristela Oliveira dos Santos. -- São  
Carlos, 2016.  
95 p.

Dissertação (Mestrado - Programa de Pós-Graduação  
em Ciências de Computação e Matemática  
Computacional) -- Instituto de Ciências Matemáticas  
e de Computação, Universidade de São Paulo, 2016.

1. OTIMIZAÇÃO COMBINATÓRIA. I. Santos, Maristela  
Oliveira dos, orient. II. Título.

**Rafael Bernardo Zanetti Cirino**

## Solution approaches for the classroom assignment problem

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Maristela Oliveira dos Santos

**USP – São Carlos**  
**July 2016**



---

# AGRADECIMENTOS

À minha orientadora, Profa. Dra. Maristela Oliveira dos Santos, por seus seus elogios e críticas, bem como por sua calma e paciência em todas as etapas do trabalho.

Ao Prof. Dr. Alysson Machado Costa, por suas lições que tangem além da academia.

Ao Prof. Dr. Alexandre Cláudio Botazzo Delbem, que contribuiu ativamente com ideias e conceitos, tanto inovadores quanto desafiantes.

Aos colegas Ms. Michel Bessani e Ms. Marcos Paulo Gonçalves Pedroso, pelas longas horas de discussão e motivação.

Aos colegas do LOt (Laboratório de Otimização) do ICMC, sempre abertos ao auxílio e debates.

À CAPES, pelo auxílio financeiro necessário para realização deste trabalho.

Pesquisa desenvolvida com utilização dos recursos computacionais do Centro de Ciências Matemáticas Aplicadas à Indústria (CeMEAI) financiados pela FAPESP.



*"(...) mas diante dos casos extremamente simples,  
assim como diante dos particularmente difíceis,  
ficavam com frequência perplexos (...)"*

**Kafka**



---

# RESUMO

Esta Dissertação aborda o Problema de Alocação de Aulas a Salas (PAAS), também conhecido como Problema de Alocação de Salas (PAS). As instituições de ensino superior, no começo de seus calendários letivos, resolvem um PAAS ao determinar os espaços a serem utilizados para as atividades didáticas. Porém, em muitas destas instituições o PAAS é ainda resolvido manualmente, gerando altas cargas de trabalho para os responsáveis. Neste trabalho, o Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP) foi tomado como caso de estudo para o PAAS. Um modelo de programação matemática inteiro é proposto e abordado por técnicas de resolução exata, metaheurísticas mono-objetivo e uma abordagem multi-objetivo. Uma estrutura de vizinhança proposta obteve resultados comparáveis à da metodologia exata, para um tempo fixo de execução. Demonstra-se que, a abordagem multi-objetivo é uma possibilidade de contornar algumas dificuldades clássicas do problema, como incertezas sobre a escolha dos pesos das métricas. Os métodos de solução propostos para o problema fornecem, aos responsáveis, bons instrumentos de auxílio à tomada de decisão para o PAAS.

**Palavras chave:** Problema de Alocação de Aulas à Salas, Problema de Alocação de Salas, Metaheurística, Multi-objetivo, Tomada de Decisão Multi-Critério, Otimização Inteira.



---

# ABSTRACT

This Dissertation addresses the Classroom Assignment Problem (CAP). All Higher Education Institutes, at the schoolyear's begin, faces a CAP to define where the classes will be taught. However, many of those still solves this problem manually, demanding high efforts from the responsible staff. In this study, the *Universidade de São Paulo's* (USP) *Instituto de Ciências Matemáticas e de Computação* (ICMC) was tackled as study case for the CAP. An Integer Programming Model is proposed and tackled by exact methods, meta-heuristics and a multi-objective approach. A novel neighborhood operator is proposed for the local search and obtains good results, even comparable to the exact method. The multi-objective approach is shown to overcome some of the classical adversity of the mono-objective approach, e.g., choosing weights to quality metric. Those CAP's proposed solution methods, gives the responsible staff a good decision making support.

**Palavras chave:** Classroom Assignment Problem, Meta-heuristics, Multi-objective Optimization, Multi-Criterion Decision Making, Integer Programming.



---

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	p. 1
<b>2</b>	<b>Revisão Bibliográfica</b>	p. 5
<b>3</b>	<b>Casos de Estudo</b>	p. 9
3.1	Modelagem com duas métricas de qualidade . . . . .	p. 10
3.2	Modelagem com cinco métricas de qualidade . . . . .	p. 13
<b>4</b>	<b>Abordagem Mono-objetivo</b>	p. 17
4.1	Heurística Construtiva . . . . .	p. 18
4.2	Busca Local . . . . .	p. 21
4.3	Busca Tabu . . . . .	p. 27
4.4	GRASP . . . . .	p. 29
4.5	Algoritmo Genético . . . . .	p. 32
4.6	Algoritmo Genético Compacto . . . . .	p. 39
<b>5</b>	<b>Abordagem Multi-objetivo</b>	p. 45
5.1	Algoritmo de Busca em Tabelas . . . . .	p. 51

<b>6</b>	<b>Testes Computacionais</b>	p. 59
6.1	Ambiente de Testes . . . . .	p. 59
6.2	Geração dos Dados . . . . .	p. 59
6.3	Resultados do Método Exato . . . . .	p. 61
6.4	Testes de Sensibilidade de Parâmetros . . . . .	p. 64
6.4.1	Apresentação dos Resultados . . . . .	p. 65
6.5	Desempenho das Abordagens Mono-objetivo . . . . .	p. 75
6.6	Desempenho dos Métodos Multi-objetivo . . . . .	p. 80
<b>7</b>	<b>Conclusões e Perspectivas</b>	p. 85
	<b>Referências Bibliográficas</b>	p. 87
<b>A</b>	<b>Algoritmos de Busca para o ABT</b>	p. 93

---

## LISTA DE TABELAS

1	Métricas de Qualidade do PAAS da Literatura . . . . .	p. 6
2	Dimensões utilizadas nas Instâncias . . . . .	p. 60
3	Instâncias utilizadas e suas características . . . . .	p. 60
4	Pesos para as 5 Métricas . . . . .	p. 61
5	Soluções obtidas com o CPLEX . . . . .	p. 62
6	Parâmetro dos Algoritmos Propostos . . . . .	p. 64
7	Discretização dos Parâmetro dos Algoritmos Propostos . . . . .	p. 64
8	Desejabilidade Composta para as combinações de parâmetros do GRASP	p. 70
9	Composite Desirability para as combinações de parâmetros do AGC . .	p. 72
10	Teste de Turkey pareado para Taxa de <i>Crossover</i> . . . . .	p. 73
11	Teste de Turkey pareado para Taxa de Mutação . . . . .	p. 74
12	Parâmetros escolhidos para cada um dos métodos mono-objetivo . . . .	p. 75
13	Parâmetros utilizados para o ABT . . . . .	p. 80



---

## LISTA DE FIGURAS

1	Ilustração da Representação por Grafo do PAAS . . . . .	p. 11
2	Exemplo do funcionamento da Busca Local . . . . .	p. 22
3	Exemplificação do movimento de Troca . . . . .	p. 23
4	Exemplificação do movimento de Alocação . . . . .	p. 23
5	Exemplificação da nova Vizinhança proposta . . . . .	p. 26
6	Exemplo do funcionamento da Busca Tabu . . . . .	p. 28
7	Representação codificada de uma solução no AG . . . . .	p. 33
8	Operador de recombinação tradicional do AG . . . . .	p. 33
9	Operador de mutação tradicional do AG . . . . .	p. 33
10	Fluxograma do AG . . . . .	p. 34
11	Codificação utilizada para o AG . . . . .	p. 35
12	Recombinação utilizando uma linha vertical . . . . .	p. 36
13	Recombinação utilizando uma linha horizontal . . . . .	p. 36
14	Processo de factibilização . . . . .	p. 37
15	Representação do modelo probabilístico $P$ . . . . .	p. 40
16	Processo de amostragem de solução para o AGC . . . . .	p. 41
17	Representação da solução $X$ no espaço de objetivos $Z$ . . . . .	p. 46

18	Ilustração do impacto da escolha dos pesos para a abordagem mono-objetivo . . . . .	p. 46
19	Relação de Dominância . . . . .	p. 47
20	Exemplo da Fronteira de Pareto . . . . .	p. 48
21	Ilustração da aproximação da FPO pelos conjuntos de solução . . . . .	p. 49
22	Ilustração de conjuntos de solução possível . . . . .	p. 49
23	Exemplos de hipervolume . . . . .	p. 50
24	Processo de Iniciação das Tabelas. . . . .	p. 53
25	GAP de otimalidade para o CPLEX com 5 minutos comparado com o limitante inferior . . . . .	p. 63
26	GAP de otimalidade para o CPLEX com 1 hora comparado com o limitante inferior . . . . .	p. 63
27	Evidência de Não Normalidade de GAP nos dados . . . . .	p. 66
28	Análise dos Resíduos para o Modelo Ajustado . . . . .	p. 67
29	Histogramas de GAP com Normais Ajustadas para o GRASP . . . . .	p. 67
30	Gráfico de Efeitos Principais de GAP . . . . .	p. 68
31	Gráfico de Probabilidade de GAP . . . . .	p. 69
32	Gráficos da Análise de Resíduos da Regressão Linear Múltipla para o GRASP . . . . .	p. 69
33	Gráfico de Efeitos Principais sobre as Médias de GAP para o GRASP <sub>T</sub> . . . . .	p. 71
34	<i>Scatterplot</i> do ensaio do AGC . . . . .	p. 72
35	Gráfico de Efeitos Principais para o Algoritmo Genético . . . . .	p. 73
36	Gráfico de Efeitos Principais para a Busca Tabu . . . . .	p. 74
37	Teste de Turkey Pareado para <i>Tabu</i> na metaheurística Busca Tabu . . . . .	p. 75
38	<i>Boxplot</i> do experimento de desempenho dos métodos mono-objetivo . . . . .	p. 76
39	<i>Boxplot</i> do desempenho dos métodos subdividido . . . . .	p. 77
40	Teste pareado de Games-Howell para o desempenho dos métodos mono-objetivo . . . . .	p. 78

41	Teste de desempenho de Dolan-Moré . . . . .	p. 79
42	Boxplot do hipervolume para as Buscas AGC, BL e BT . . . . .	p. 81
43	<i>Boxplot</i> da Busca AGC dividida em categorias . . . . .	p. 82
44	Teste t pareado para a média dos hipervolumes das buscas . . . . .	p. 83



---

# LISTA DE ABREVIATURAS E SIGLAS

ABT	Algoritmo de Busca em Tabelas
AED	Algoritmos de Estimação de Distribuição
AG	Algoritmo Genético
AGC	Algoritmo Genético Compacto
ANOVA	Analysis of Variance
BT	Busca Tabu
CAP	Classroom Assingment Problem
FP	Fronteira de Pareto
FPO	Fronteira de Pareto Ótima
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
ICMC	Instituto de Ciências Matemáticas e de Computação
LT	Lista Tabu
MEAN	Multiobjective Evolutionary Algorithm with Node-depth encoding
PAAS	Problema de Alocação de Aulas à Salas
PAS	Problema de Alocação de Salas
RLC	<i>Restricted Candidate List</i>



---

---

# CAPÍTULO 1

---

## INTRODUÇÃO

Este trabalho trata o Problema de Alocação de Aulas à Salas (PAAS) e discute diferentes abordagens para resolvê-lo. O PAAS consiste em encontrar a melhor forma de distribuir aulas (eventos) nas salas (espaços), atendendo à exigências dos decisores locais. Este problema pode ser abordado diretamente ou como parte constituinte dos Problemas de Programação de Cursos Universitários.

O Problema de Programação de Cursos Universitários é um problema de alocação de recursos que deve definir os horários, docentes e salas para a realização das aulas, atendendo à demanda didática das Universidades. O PAAS portanto, é uma parte deste problema, pois considera que os horários das aulas já definidos e alocam-se somente as salas para estas aulas. O PAAS pode ser resolvido separadamente ou como parte do Problema de Programação de Cursos Universitários (PHILLIPS et al., 2015).

Nesta dissertação será considerado o PAAS em instituições de ensino superior como um problema específico, ou seja, os horários de início e término de cada aula e os agentes discentes e docentes já estão definidos à priori. Considerou-se que para estas instituições existam cursos estabelecidos que possuem sua carga didática predefinida, assim, para que um aluno possa se graduar como Engenheiro Civil, por exemplo, deverá cursar algumas aulas específicas, i.e., obrigatórias para obtenção deste título.

Deve-se portanto destacar alguns conceitos que serão utilizados ao longo da dissertação:

- **Aula:** Um evento semanal com horário de início, horário de término, docente e alunos predefinidos;
- **Sala:** Um espaço apropriado para a realização das aulas;
- **Turma:** Um conjunto de alunos com um professor que possui uma ou mais aulas semanais, por exemplo, uma turma de Introdução à Didática que possua 2 aulas semanais;
- **Currículo:** Um conjunto de aulas assistidas por um grupo discente predefinido. Por exemplo, as aulas obrigatórias do terceiro período do curso de Matemática;
- **Recurso:** Os recursos são elementos associados às aulas na forma de exigências didáticas ou ergonômicas, enquanto são associadas às salas na forma de disponibilidade. Por exemplo, acessibilidade, projetor, mesa de desenho, etc.

Estes elementos estão presentes em qualquer trabalho relativo ao PAAS, pois ao menos o número de assentos da sala e o número de alunos na aula podem ser vistos como recursos e o agrupamento de aulas em grupos curriculares ou turmas é uma prática comumente adotada. Nesta dissertação, os recursos serão tratados de forma impeditiva, ou seja, caso a sala não possua o recurso exigido para uma aula, sua alocação ficará inviabilizada.

Existem diversas formulações para o problema na literatura, pois em cada local em que o trabalho é considerado, cria-se uma nova formulação, dado que os diferentes decisores possuem distintas medidas de qualidade que almejam ao abordar o problema. Porém existem algumas características que são intrínsecas ao PAAS, que são resumidas em três restrições fortes:

1. Todas as aulas devem ser alocadas em alguma sala;
2. As aulas não podem ter intersecção de horário nas salas, ou seja, duas aulas com sobreposição de horário não podem ser simultaneamente alocadas na mesma sala;
3. As aulas somente poderão ser alocadas em salas que atendam suas demandas de recursos e capacidade para acomodar todos os participantes.

Nesta dissertação foram propostas algumas abordagens para resolver o problema para o caso de estudo no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo campus São Carlos. Um modelo matemático é proposto e abordado pela ferramenta IBM ILOG CPLEX.

As abordagens mono-objetivo e multi-objetivo são consideradas e alguns métodos de resolução heurísticos e metaheurísticos foram propostos para o problema. Foram realizados experimentos estatísticos para análise da escolha dos parâmetros e de desempenho destes métodos.

Análises de desempenho indicam que os métodos propostos são robustos, porém divergem na qualidade das soluções obtidas. Para os métodos multi-objetivo as soluções são analisadas por meio de uma medida unária bem estabelecida na literatura, o hipervolume. São realizadas algumas análises e alguns comentários sobre a relação entre as abordagens mono e multi objetivo são tecidos.

Esta dissertação está dividida em capítulos. O Capítulo 2 traz uma revisão bibliográfica do problema. No Capítulo 3 discute-se um estudo de caso e alguns exemplos de formulação por meio de modelagem matemática. No Capítulo 4, são apresentadas as diferentes abordagens mono-objetivo estudadas para a resolução do problema, enquanto o Capítulo 5 apresenta uma abordagem multi-objetivo.

O Capítulo 6, discorre sobre a metodologia de testes e os resultados obtidos por meio das abordagens propostas. O Capítulo 7 apresenta as conclusões, aplicações, perspectivas e ideias para trabalhos futuros.



---

---

## CAPÍTULO 2

---

### REVISÃO BIBLIOGRÁFICA

O PAAS é um problema que atinge várias instituições de ensino superior ao redor do mundo durante o começo de seus períodos letivos. Por exemplo, a Universidade de Auckland (WATERER, 1995) e (PHILLIPS et al., 2015), a Universidade de Purdue (MOONEY et al., 1996) e a Universidade de Sfax (ELLOUMI et al., 2014). Existem outros trabalhos que abordam o problema como parte integrante do Problema de Programação de Cursos Universitários, que foge do escopo desta dissertação, para mais informações ver Pillay (2014).

Várias Instituições de Ensino Superior do Brasil abordam o problema. Alguns exemplos que foram analisados para esta dissertação são os trabalhos de (SOUZA; SCARPIN, 2014) na Universidade Federal do Paraná; (SOUZA et al., 2002a), (SOUZA et al., 2002b), (SILVA; SILVA, 2009) e (PRADO; SOUZA, 2014) na Universidade Federal de Ouro Preto; (SUBRAMANIAN et al., 2011) na Universidade Federal da Paraíba; (KRIPKA; KRIPKA, 2010) na Universidade de Passo Fundo; (CONSTANTINO et al., 2010) na Universidade Estadual de Maringá.

Os primeiros trabalhos sobre o PAAS remontam aos anos 70, com o trabalho de Dyer e Mulvey (1976) que mais tarde é estendido para o primeiro modelo de alocação de aulas á salas Mulvey et al. (1982). A sua complexidade foi demonstrada no trabalho de Carter e Tovey (1992) e é verificado que o PAAS é um problema NP-Completo, redutível ao problema de k-coloração de grafo.

Tabela 1: Métricas de Qualidade do PAAS da Literatura

Publicação	Métricas				
	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$
Mulvey et al. (1982)				•	
Waterer (1995)				•	•
Martinez-Alfaro e Flores-Teran (1998)	•			•	
Souza et al. (2002a)		•	•	•	
Souza et al. (2002b)	•	•	•	•	
Nascimento et al. (2005)		•	•		
Silva e Silva (2009)		•	•	•	
Constantino et al. (2010)	•	•			•
Kripka e Kripka (2010)	•				
Subramanian et al. (2011)		•	•		•
Cirino et al. (2013)		•		•	
Prado e Souza (2014)	•	•	•	•	•
Souza e Scarpin (2014)		•	•	•	•
Elloumi et al. (2014)				•	
Phillips et al. (2015)		•		•	•
Esta Dissertação	•	•	•	•	•

A Tabela 1 sumariza as diferentes métricas de qualidade utilizadas na literatura. Elas foram definidas de forma a englobar as diferentes maneiras como são apresentadas na literatura, mas preservando o fator essencial de sua medição. Por exemplo, para a métrica **Redução do deslocamento** ( $M_1$ ) foram considerados tanto os casos de deslocamento de docente (PRADO; SOUZA, 2014) quanto o deslocamento de discentes (KRIPKA; KRIPKA, 2010) e que na prática transcrevem-se em verificar se um conjunto previamente estabelecido de aulas está disperso segundo as métricas estabelecidas, normalmente a distância.

Com a métrica de **Estabilidade de Sala** ( $M_2$ ), tenta-se manter em determinadas salas, ou conjuntos de salas, um conjunto previamente especificado de aulas. Assim, trabalhos que consideram que aulas de um mesmo **Currículo** devem permanecer em um mesmo bloco didático (SUBRAMANIAN et al., 2011) ou trabalhos que consideram que aulas de uma mesma **Turma** devem ser alocadas na mesma sala (PHILLIPS et al., 2015) constituem a mesma categoria.

Ressalta-se que a métrica **Restrição de Utilização de Salas** ( $M_3$ ) é considerada tanto na sua forma parcial, i.e. há alguns horários indesejáveis para sua utilização ou evita-se a sub-utilização dos recursos das salas (SUBRAMANIAN et al., 2011), até sua forma total, i.e. é indesejável a utilização da sala em qualquer horário (NASCIMENTO et al., 2005), (CIRINO et al., 2015).

Para a métrica **Melhor utilização da capacidade da Sala** ( $M_4$ ) é relativas ao

número de assentos livres nas salas após a alocação de uma aula, normalmente tentando reduzir este número. Enquanto a métrica **Atendimento de Preferências por Salas** ( $M_5$ ) inclui tanto as preferências de docentes por salas (PRADO; SOUZA, 2014) quanto preferências de discentes (PHILLIPS et al., 2015).

As abordagens de solução para o problema distinguem-se muito na literatura, destacando-se o uso da meta-heurística Recozimento Simulado (*Simulated Annealing*) em (KRIPKA; KRIPKA, 2010) (SOUZA et al., 2002a) (MARTINEZ-ALFARO; FLORES-TERAN, 1998) (NASCIMENTO et al., 2005); da meta-heurística Busca Tabu (SOUZA et al., 2002a) (SUBRAMANIAN et al., 2011) (SOUZA et al., 2004); além de abordagens distintas como Coloração de Grafos (SILVA; SILVA, 2009), heurísticas construtivas (LOPES; SCHOEFFEL, 2002) e técnicas de geração de colunas para aproximação da solução do modelo inteiro (CONSTANTINO et al., 2010).

Finalmente, há de se considerar que o PAAS não é um problema único, pois matematicamente é o mesmo problema de alocar quartos de hotel, alocação de navios em docas, alocação de portões de embarques para aeronaves e planejamento de tarefas para máquinas sem capacitação (WATERER, 1995) (CARTER; TOVEY, 1992).



---

---

## CAPÍTULO 3

---

### CASOS DE ESTUDO

O PAAS tem formulação matemática simples para gerar soluções factíveis, ou seja, soluções que atendam à demanda das restrições fortes do problema: completude de alocação, não sobreposição de horários e atendimento de requisitos. Porém dada a alta dependência das métricas de qualidade com o local em que o problema é resolvido, os estudos de caso irão desenvolver modelos de forma a atender estes requisitos (ver Tabela 1 no Capítulo 2).

Para esta dissertação foi estudado o caso do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo - Campus São Carlos (ICMC). O ICMC possui seis cursos de graduação e é co-responsável por mais dois, além de oferecer quatro programas de pós-graduação e possuir vinte e três salas didáticas dispostas em 4 blocos de aula. Somente as aulas que utilizam os espaço físico do ICMC são consideradas para o trabalho.

Os responsáveis por resolver o problema no ICMC, chamados doravante de **decisores**, resolvem o problema manualmente e utilizam soluções históricas como base principal. Esta metodologia mostrou-se inadequada para alguns estudos de expansão, pois adicionar uma aula nova ou mudar o horário de aulas antigas, alteraria a solução histórica a ponto de ser necessária uma nova solução - solução esta que poderia demorar várias horas para ser obtida.

No PAAS existe uma variabilidade grande quanto às métricas de qualidade das so-

luções e cada local estabelece seus critérios. Por vezes esses critérios não são explícitos, pois os decisores normalmente atribuem qualidade à solução por meio de experiências anteriores ou respostas dos discentes ou docentes. Assim por meio de entrevistas com os decisores foi-se estabelecida o primeiro modelo para o ICMC com somente duas métricas de qualidade (Seção 3.1).

Os resultados obtidos com o primeiro modelo para o ICMC serviram como base para compreender as outras formas de medir qualidade que não haviam, até então, sido formuladas matematicamente. Tendo em vista que o conhecimento adquirido para a resolução do problema é totalmente empírico, dificultando portanto a transcrição dos reais intuitos dos decisores. Por meio do processo de identificação dos problemas na solução do primeiro modelo e de novas reuniões com os decisores, ficou estabelecido um novo modelo, este agora com cinco métricas, para o PAAS no ICMC (Seção 3.2).

### 3.1 Modelagem com duas métricas de qualidade

Para a formulação do PAAS com duas métricas no ICMC as aulas serão tratadas independente da grade horária. A grade horária servirá somente para estabelecer as relações pertinentes entre as aulas e salas, por exemplo, definir as relações de sobreposição de horário entre as aulas. Elimina-se a necessidade de tratar os *slots* de horário nas variáveis de decisão.

O tempo de uma semana é discretizado por meio de *slots* de horário e cada aula deverá estar associada a um ou mais destes *slots* consecutivos. Destaca-se que estes *slots* podem possuir durações diferentes, bastando cobrir todos os horários da semana. A relação entre as aulas das turmas e os *slots* de horário é estabelecida por meio de uma matriz binária  $\rho$ , sendo um dado de entrada do problema.

Nesse modelo cada aula será um envelopamento de seus dados, ou um metadado, contendo horário de início, horário de término, docente, número de inscritos, recursos requeridos e turma. Os conjuntos de aula em sobreposição podem ser extraídos dos dados e adicionados nas restrições. Optou-se por modelar o problema alocando aulas a salas, mantendo-se as três restrições fortes, ou seja, completude da solução, não sobreposição de horários e atendimento de requisitos de recursos.

Pode-se representar parcialmente o PAAS por meio de um grafo, atribuindo cada aula à um vértice deste grafo e adicionando arestas para cada par de aula com sobreposição de pelo menos 1 *slot* de horário. Obtém-se então a matriz de adjacências  $\Theta$  (matriz

de conflitos de aulas) e modela-se a restrição de não-sobreposição de horários como em Ferland e Roy (1985). A Figura 1 ilustra um grafo representando parcialmente um PAAS. No exemplo, as aulas A7 e A6 possuem sobreposição de horário.

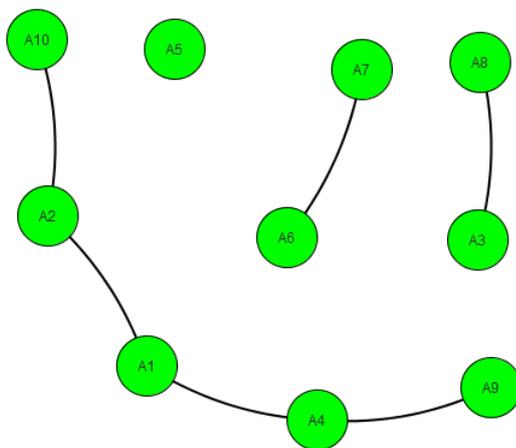


Figura 1: Ilustração da Representação por Grafo do PAAS

Os recursos considerados para este modelo são: sala de aula comum, quadro grande, projetor, laboratório de informática e laboratório de matemática. Para cada turma os docentes determinam os requerimentos didáticos de recursos, estabelecendo uma relação binária entre as turmas e os recursos (matriz  $\tau$ ) que é um dado de entrada do problema. Para cada sala os decisores sabem quais os recursos estão associados (matriz binária  $\sigma$ ) que é também dado de entrada.

Para modelar a restrição de atendimento de requisitos foi proposta uma matriz binária  $\eta$  relacionando aulas e salas. Para construir esta matriz utiliza-se o número de alunos na turma ( $TAM_t$ ), a capacidade das salas ( $CAP_s$ ) e as matrizes  $\tau$  e  $\sigma$ . Estabelece-se que uma aula  $a$  poderá esta alocada na sala  $s$  caso esta possua número de assentos igual ou superior ao número de alunos e disponibilize, ao menos, todos os recursos requeridos pela aula. Portanto  $\eta_{as}$  será 1 para todas as salas  $s$  que podem alocar a aula  $a$  e 0 para as demais.

Nesta etapa inicial do desenvolvimento do modelo, os decisores do ICMC salientaram somente duas métricas de qualidade para aferir uma solução obtida. Estas duas métricas são: utilizar salas com capacidade o mais próxima possível da quantidade de alunos da turma, ou seja, reduzir o número de assentos vazios em cada sala; para uma ou mais aulas que sejam da mesma turma, tentar alocá-las na mesma sala, ou seja, reduzir o número de trocas de sala das turmas.

A modelagem da primeira métrica é feita com o uso de uma matriz  $USO$ , que associa

ao par aula-sala um custo relativo caso a aula seja alocada na sala. Com o intuito de reduzir o número de assentos livres nas salas, define-se o  $USO_{as} \doteq 100 \left(1 - \frac{TAM_{t_a}}{CAP_s}\right)$ , em que  $TAM_{t_a}$  é o número de alunos inscritos na aula. O  $USO_{as}$  portanto irá ser um valor percentual de assentos vazios na sala  $s$  caso a aula  $a$  seja alocada na mesma.

Os dados disponibilizados pelos decisores sobre as turmas, recursos e salas passam por pré-processamento, para reduzir a quantidade de informação e extrair as relações importante entre aulas e salas. Finalmente cabe ao decisor a escolha apropriada dos parâmetros  $\alpha$  e  $\beta$  que ponderam a função objetivo, definindo comensurabilidade entre as métricas. O modelo do PAAS para o ICMC com duas métricas é então proposto:

#### Conjuntos e Índices

- $H$  Conjunto de *slots* de horário ( $h \in \{1, \dots, |H|\}$ );
- $T$  Conjunto de turmas ( $t \in \{1, \dots, |T|\}$ );
- $S$  Conjunto de salas ( $s \in \{1, \dots, |S|\}$ );
- $R$  Conjunto de recursos ( $r \in \{1, \dots, |R|\}$ ).

#### Dados de Entrada

- $\rho$   $\rho_{th} = 1$  se turma  $t \in T$  tem aula no *slot* de horário  $h \in H$ , 0 c.c.;
- $\sigma$   $\sigma_{sr} = 1$  se sala  $s \in S$  possui recurso  $r \in R$ , 0 c.c.;
- $\tau$   $\tau_{tr} = 1$  se turma  $t \in T$  requer recurso  $r \in R$ , 0 c.c.;
- CAP**  $CAP_s =$  número de assentos na sala  $s \in S$ ;
- TAM**  $TAM_t =$  número de alunos inscritos na turma  $t \in T$ ;

#### Dados de Pré-processamento

- $A$  Conjunto de Aulas ( $a \in \{1, \dots, |A|\}$ );
- $A^t$  Subconjunto das Aulas  $a \in A$  da turma  $t \in T$
- $\eta$   $\eta_{as} = 1$  se sala  $s \in S$  pode alocar a aula  $a \in A$ , 0 c.c.;
- $\Theta$   $\Theta_{aa'} = 1$  se as aulas  $a, a' \in A$  possuem sobreposição de horário, 0 c.c.;
- USO**  $USO_{as} =$  custo de alocar aula  $a$  na sala  $s$ .

#### Variáveis de Decisão

- $x_{as}$   $x_{as} = 1$  se aula  $a \in A$  é alocada na sala  $s \in S$ , 0 c.c.;
- $y_t$   $y_t =$  número de trocas da turma  $t \in T$ .

## Formulação do ICMC com 2 métricas

$$\min \quad F = \alpha \cdot \left( \sum_{s \in S} \sum_{a \in A} USO_{as} \cdot x_{as} \right) + \beta \cdot \left( \sum_{t \in T} y_t \right) \quad (3.1)$$

$$\text{sujeito a:} \quad \sum_{s \in S} x_{as} = 1 \quad a \in A \quad (3.2)$$

$$x_{as} \leq \eta_{sa} \quad s \in S \text{ e } a \in A \quad (3.3)$$

$$x_{as} + x_{a's} \leq 1 \quad s \in S \text{ e } a, a' \in A \mid \Theta_{aa'} = 1 \quad (3.4)$$

$$\sum_{n=1}^{|A^t|} x_{a_n s_n} \leq 1 + y_t \quad t \in T, a_n \in A^t \text{ e } \{s_1, \dots, s_{|A^t|}\} \subset S \quad (3.5)$$

$$x_{as} \in \{0, 1\} \quad a \in A, s \in S \quad (3.6)$$

$$y_t \in \mathbb{Z}_+ \quad t \in T \quad (3.7)$$

A Função Objetivo (3.1) minimiza o percentual de assentos livres nas salas (termo multiplicado por  $\alpha$ ) e o número de trocas de salas das turmas (termo multiplicado por  $\beta$ ).

O conjunto de restrições (3.2) assegura que todas as aulas devem ser alocadas em uma, e somente uma, sala. As restrições (3.3) garantem que só serão alocadas aulas em salas que cumpram as exigências de espaço e recursos. As restrições (3.4) impedem que duas aulas com conflito de horário sejam alocadas na mesma sala, enquanto as restrições (3.5) irão medir a quantidade de trocas de salas das turmas. O conjunto de restrições (3.6) e (3.7) são de domínio de variáveis.

## 3.2 Modelagem com cinco métricas de qualidade

As soluções obtidas para o modelo com duas métricas para o ICMC foram conferidas juntamente com os decisores do PAAS e alguns problemas na solução foram levantadas pelos mesmos. Medidas de qualidade que eram consideradas implicitamente por eles não haviam sido transpostas para o modelo até então, o que levou a soluções de baixa qualidade segundo as métricas não consideradas.

O novo modelo utiliza as mesmas premissas do modelo anterior, com a adição de novos dados de entrada e de pré-processamento. O conjunto de currículos, adicionado para este modelo, diz respeito a um conjunto de disciplinas que um determinado grupo discente, por conta do curso escolhido e ano de ingresso, deverá fazer. Associa-se então para cada turma  $t \in T$ , os currículos  $\{c_1, \dots, c_n\} \subset C$ , por meio da matriz binária  $\Upsilon$ . Uma mesma turma pode estar associada a mais de um currículo.

As soluções do primeiro modelo utilizavam demasiadamente as salas menores do ICMC para turmas pequenas, porém estas salas são geralmente utilizadas para defesas e seu uso para aulas semanais deve ser evitado. Existe, portanto, um conjunto de salas que devem permanecer preferencialmente vazias ao longo do semestre. Este conjunto de salas será dado pelo vetor **PVAZ**, em que o valor 1 indica que a sala  $s$  é preferencialmente vazia.

Um outro problema detectado na solução anterior era a baixa alocação de aulas de alguns currículos específicos à salas predefinidas. Portanto outra métrica definida é preservar uma relação pré-estabelecida entre currículos e salas de aula. Esta relação será dada pela matriz de preterências **PRET**, que associará a cada par currículo-sala um valor entre 0 e 10, em que 0 são as salas mais indicadas para o currículo e 10 são as salas menos indicadas. À guisa de exemplo, os ingressantes da universidade tem preferência por utilizar as salas de aula mais próximas do prédio da administração, portanto o valor de **PRET** deste currículo será 0 para estas salas.

Os decisores indicaram então a última das métricas que ainda não havia sido considerada, a redução do deslocamento dos alunos. Para modelar esta métrica requer do decisor os dados sobre as distâncias entre as salas (matriz **DIS**). Reduz-se então o deslocamento de cada um dos currículos, i.e., as aulas de um determinado currículo devem ser alocadas em salas próximas, segundo a matriz de distância **DIS**.

Destaca-se porém que as duas métricas utilizadas no modelo da Seção 3.1 são mantidas, acrescentando-se outras três métricas, totalizando as cinco métricas para o novo modelo do ICMC:

1. Alocar aulas em salas que tenham a capacidade mais próxima do número de alunos;
2. Evitar trocas de sala para aulas de uma mesma disciplina;
3. Reduzir a distância percorrida por um currículo, considerando todas as suas aulas;
4. Evitar a utilização de um determinado conjunto de salas de aula;
5. Atender às preferências pré-estabelecidas entre currículo e salas de aula.

Analogamente ao modelo com duas métricas cabe ao decisor estipular os pesos dos parâmetros de ponderação da função objetivo, i.e., estabelecer a comensurabilidade entre as métricas a serem minimizadas. Portanto, cada uma das métricas estará associada a um peso a ser definido pelo decisor:  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ ,  $\epsilon$ , respectivamente.

## Conjuntos e Índices

$H$	Conjunto de <i>slots</i> de horário ( $h \in \{1, \dots,  H \}$ );
$T$	Conjunto de turmas ( $t \in \{1, \dots,  T \}$ );
$S$	Conjunto de salas ( $s \in \{1, \dots,  S \}$ );
$R$	Conjunto de recursos ( $r \in \{1, \dots,  R \}$ );
$C$	Conjunto de currículos ( $c \in \{1, \dots,  C \}$ ).

## Dados de Entrada

$\rho$	$\rho_{th} = 1$ se turma $t \in T$ tem aula no <i>slot</i> de horário $h \in H$ , 0 c.c.;
$\sigma$	$\sigma_{sr} = 1$ se sala $s \in S$ possui recurso $r \in R$ , 0 c.c.;
$\tau$	$\tau_{tr} = 1$ se turma $t \in T$ requer recurso $r \in R$ , 0 c.c.;
$\Upsilon$	$\Upsilon_{tc} = 1$ se turma $t \in T$ pertence ao currículo $c \in C$ , 0 c.c.;
<b>CAP</b>	$CAP_s =$ número de assentos na sala $s \in S$ ;
<b>TAM</b>	$TAM_t =$ número de alunos inscritos na turma $t \in T$ ;
<b>DIS</b>	$DIS_{ss'} =$ distância entre as salas $s \in S$ e $s' \in S$ ;
<b>PVAZ</b>	$PVAZ_s = 1$ se sala $s \in S$ é preferencialmente vazia, 0 c.c.;
<b>PRET</b>	$PRET_{cs} =$ preterência do currículo $c \in C$ pela sala $s \in S$ .

## Dados de Preprocessamento

$A$	Conjunto de Aulas ( $a \in \{1, \dots,  A \}$ );
$A^t$	Subconjunto das Aulas $a \in A$ da turma $t \in T$ ;
$\Lambda^c$	Subconjunto de Aulas $a \in A$ do currículo $c \in C$ ;
$\eta$	$\eta_{as} = 1$ se sala $s \in S$ pode alocar a aula $a \in A$ , 0 c.c.;
$\Theta$	$\Theta_{aa'} = 1$ se as aulas $a, a' \in A$ possuem sobreposição de horário, 0 c.c.;
<b>USO</b>	$USO_{as} =$ custo de alocar aula $a$ na sala $s$ .

## Variáveis de Decisão

$x_{as}$	$x_{as} = 1$ se aula $a \in A$ é alocada na sala $s \in S$ , 0 c.c.;
$y_t$	$y_t =$ número de trocas da turma $t \in T$ ;
$w_{cs}$	$w_{cs} = 1$ se currículo $c \in C$ tem ao menos 1 aula na sala $s \in S$ , 0 c.c.;
$v_{css'}$	$v_{css'} = 1$ se currículo $c \in C$ tem ao menos 1 aula nas salas $s, s' \in S$ , 0 c.c.

## Formulação do ICMC com 5 métricas

$$\begin{aligned} \min \quad F = & \alpha \cdot \left( \sum_{s \in S} \sum_{a \in A} USO_{as} x_{as} \right) + \beta \cdot \left( \sum_{t \in T} y_t \right) + \gamma \cdot \left( \sum_{c \in C} \sum_{s \in S} \sum_{s' \in S \setminus \{s\}} DIS_{ss'} v_{css'} \right) \\ & + \delta \cdot \left( \sum_{s \in S} \sum_{a \in A} PVAZ_s x_{as} \right) + \epsilon \cdot \left( \sum_{c \in C} \sum_{s \in S} PRET_{cs} w_{cs} \right) \end{aligned} \quad (3.8)$$

$$\text{sujeito a:} \quad \sum_{s \in S} x_{sa} = 1 \quad a \in A \quad (3.9)$$

$$x_{as} \leq \eta_{sa} \quad s \in S \text{ e } a \in A \quad (3.10)$$

$$x_{as} + x_{a's} \leq 1 \quad s \in S \text{ e } a, a' \in A \mid \Theta_{aa'} = 1 \quad (3.11)$$

$$\sum_{n=1}^{|A^t|} x_{a_n s_n} \leq 1 + y_t \quad t \in T, a_n \in A^t \text{ e } \{s_1, \dots, s_{|A^t|}\} \subset S \quad (3.12)$$

$$x_{as} \leq w_{cs} \quad s \in S, c \in C \text{ e } a \in \Lambda^c \quad (3.13)$$

$$2v_{css'} \leq w_{cs} + w_{cs'} \quad c \in C, s, s' \in S \quad (3.14)$$

$$v_{css'} \geq w_{cs} + w_{cs'} - 1 \quad c \in C, s, s' \in S \quad (3.15)$$

$$x_{as} \in \{0, 1\} \quad a \in A, s \in S \quad (3.16)$$

$$y_t \in \mathbb{Z}_+ \quad t \in T \quad (3.17)$$

$$v_{css'} \in \{0, 1\} \quad c \in C, s, s' \in S \quad (3.18)$$

$$w_{cs} \in \{0, 1\} \quad c \in C, s \in S \quad (3.19)$$

A Função Objetivo (3.8) atende portanto as 5 métricas propostas. Os parâmetros  $\alpha$  e  $\beta$  cumprem a mesma função que cumpriam em (3.1), ponderando as métricas de uso e troca de sala respectivamente. A métrica de redução do deslocamento de currículo é ponderada pelo parâmetro  $\gamma$ , o parâmetro  $\delta$  será ponderação da métrica de restrição de salas preferencialmente vazias e a métrica de atendimento de preferências de salas é ponderada pelo parâmetro  $\epsilon$  (minimizando a preterência).

As restrições (3.9)–(3.12) são análogas às do modelo anterior. As restrições (3.13) fazem a ligação entre as variáveis  $x_{as}$  e  $w_{cs}$  e os conjuntos de restrições (3.14) e (3.15) fazem a ligação das variáveis  $w_{cs}$  e  $v_{css'}$ , por meio de uma operação lógica e, ou seja, caso  $w_{cs} = w_{cs'} = 1$  então  $v_{css'} = 1$ . Finalmente as restrições (3.16)–(3.19) são domínios de variáveis.

---

---

## CAPÍTULO 4

---

### ABORDAGEM MONO-OBJETIVO

O PAAS é caracterizado por possuir muitas métricas atribuídas, pois em uma alocação de aulas a salas, existem muitos agentes interagindo - discentes, docentes, serviço de limpeza - e as suas necessidades, por vezes, são conflitantes. Normalmente os decisores tentam abordagens que contemplem as peculiaridades de cada um dos envolvidos e quando tratados matematicamente, geram modelos com muitas restrições e difíceis de serem tratados.

Ao modelar diversas métricas de qualidade, uma abordagem possível é estipular pesos  $w_i$  para cada métrica  $f_i$  e minimizar a função ponderada  $F = \sum w_i f_i$ . Estes pesos tornarão comensuráveis as diferenças entre as métricas e gerarão, portanto, uma ordem de importância para estas.

A principal vantagem de minimizar uma função ponderada é o arcabouço teórico disponível para resolução do modelo, tanto de forma exata quanto métodos aproximados. Há, porém, a desvantagem da dependência da escolha dos pesos para medir as diferenças entre as grandezas que quantificam as métricas. Por exemplo, para o caso do ICMC (Capítulo 3) pode-se levantar a questão: Quantos metros de deslocamento de um perfil equivalem a uma troca de sala de uma turma?

Para todos os métodos mono-objetivo foi utilizada a definição de solução factível do PAAS dada na Definição 1. Uma instância do PAAS é o conjunto de dados de entrada e de pré-processamento discutidos no Capítulo 3. Os conjuntos  $C^a$  e  $S^a$  são inerentes aos

dados que caracterizam a instância e portanto estão bem definidos.

**Definição 1.** *Seja  $A$  um conjunto de aulas e  $S$  um conjunto de salas de uma instância do PAAS. Sejam ainda os conjuntos*

$$C^a \doteq \{a_i \in A \setminus \{a\} \mid \exists \text{ sobreposição de horário entre } a \text{ e } a_i\}$$

$$S^a \doteq \{s \in S \mid a \text{ pode ser alocada em } s \in S\}$$

Uma **solução factível do PAAS** será um mapa  $X : A \rightarrow S$ , tal que  $X(a) = s$ , que atenderá as seguintes duas propriedades:

1.  $\forall a \in A \implies \exists! s \in S^a$  tal que  $X(a) = s$ ;
2.  $\forall a \in A$  e  $\forall a_i \in C^a \implies X(a) \neq X(a_i)$ ;

## 4.1 Heurística Construtiva

Heurísticas construtivas são procedimentos incrementais para resolução de problemas, largamente utilizadas para diversos tipos de problema ou como parte constituinte de outros procedimentos. As heurísticas construtivas partem de um conjunto vazio e a cada iteração adicionam elementos um a um que constituirão a solução. Uma vez associado um elemento ele não se desassocia no processo de construção.

A heurística construtiva proposta é baseada no procedimento utilizado para a resolução manual do PAAS pelos decisores do ICMC. No ICMC se inicia com uma solução vazia e em cada passo associa-se uma sala para uma ou mais aulas, continuando até que todas as aulas tenham sido alocadas.

Utilizando-se da definição de solução factível do PAAS (Definição 1), a heurística construtiva proposta parte de um mapa  $X$  vazio. O conjunto  $A$  de aulas é ordenado de forma decrescente pelo número de inscritos e para cada aula  $a$  é escolhida uma sala  $s$ , fazendo  $X(a) = s$ . A escolha da sala é feita utilizando-se da função  $\Delta_a(X, s)$  que define para cada par  $(a, s)$  o impacto na função objetivo da escolha da sala  $s$  para a aula  $a$ , dado as alocações já realizadas em  $X$ .

O cálculo de  $\Delta_a(X, s)$  avalia o impacto na função objetivo ao alocar a sala  $s$  para a aula  $a$ , dado as alocações  $X$  já realizadas, conforme é ilustrado pelo o Algoritmo 4.1. Assim, para cada métrica  $i$  é avaliada a variação em  $i$  da escolha da aula, os valores

**Algoritmo 4.1:** Pseudocódigo do Cálculo da Função  $\Delta_a(X, s)$ 


---

**Entrada:** Aula  $a$ , Sala  $s$ , Mapa Atual  $X$ , Matrizes  $USO_{as}$ ,  $DIS_{ss'}$ ,  $PVAZ_s$  e  $PRET_{cs}$   
**Saída:** Valor de  $\Delta_a(X, s)$

- 1 Seja  $t_a$  a turma da qual a aula  $a$  faz parte;
- 2 Seja  $c_a = \{c^1, \dots, c^k\}$  os currículos dos quais a aula  $a$  faz parte;
- 3 Faça  $\Delta_a^1(X, s) = \alpha \cdot USO_{as}$  ;
- 4 se  $|A^{t_a}| = 1$  então
- 5 | Faça  $\Delta_a^2(X, s) = 0$  ;
- 6 senão
- 7 | Seja a variável temporária  $\lambda = 0$ ;
- 8 | para cada  $a_i$  da turma  $t_a$  diferente de  $a$  faça
- 9 | | se  $\exists X(a_i) \neq s$  então
- 10 | | | Faça  $\lambda = \lambda + \beta$ ;
- 11 | | fim
- 12 | fim
- 13 | Faça  $\Delta_a^2(X, s) = \lambda$ ;
- 14 fim
- 15 Seja a variável temporária  $\mu = 0$ ;
- 16 para cada  $c^i \in c_a$  faça
- 17 | se  $w_{c^i s} = 0$  então
- 18 | | para cada  $s' \in S \setminus \{s\}$  faça
- 19 | | | se  $w_{c^i s'} = 1$  então
- 20 | | | | Faça  $\mu = \mu + \gamma \cdot DIS_{ss'}$ ;
- 21 | | | fim
- 22 | | fim
- 23 | fim
- 24 fim
- 25 Faça  $\Delta_a^3(X, s) = \mu$  ;
- 26 se  $PVAZ_s = 1$  então
- 27 | Faça  $\Delta_a^4(X, s) = \delta$  ;
- 28 senão
- 29 | Faça  $\Delta_a^4(X, s) = 0$  ;
- 30 fim
- 31 Seja a variável temporária  $\pi = 0$ ;
- 32 para cada  $c^i \in c_a$  faça
- 33 | se  $w_{c^i s} = 0$  então
- 34 | | Faça  $\pi = \pi + \epsilon \cdot PRET_{c^i s}$ ;
- 35 | fim
- 36 fim
- 37 Faça  $\Delta_a^5(X, s) = \pi$  ;
- 38 Faça  $\Delta_a(X, s) = \sum_{j=1}^5 \Delta_a^j(X, s)$  ;
- 39 retorna  $\Delta_a(X, s)$

---

$\Delta_a^i(X, s)$  nas linhas 3, 5, 13, 25, 27, 29, 37 do Algoritmo 4.1.  $\Delta_a(X, s)$  será a soma das variações em cada métrica (linha 38).

As variações para a primeira, segunda e quarta métricas são obtidas somente por

aplicação de funções lógicas. As variações na terceira e quinta métrica são calculadas utilizando uma matriz binária auxiliar  $\mathbf{w} \in \{0, 1\}^{|C| \times |S|}$ , sendo  $C$  o conjunto de currículos da instância e  $S$  o conjunto de salas. Esta matriz  $\mathbf{w}$  é iniciada com todos os valores iguais a zero e é atualizada na execução da heurística construtiva, linha 11 do Algoritmo 4.2.

Procura se obter somente soluções factíveis com a heurística. O cálculo de  $\Delta_a(X, s)$  não considera a factibilidade de alocar a aula  $a$  na sala  $s$ , portanto a factibilidade deve ser verificada antes. Para tanto define-se, pela expressão (4.1), o conjunto  $S_1^a$  das salas possíveis para a aula  $a$ , i.e., as salas que possuem os recursos e a capacidade suficiente para alocar a aula  $a$ . A matriz  $\boldsymbol{\eta}$  é obtida no preprocessamento dos dados (Capítulo 3).

$$S_1^a \doteq \{s \in S \mid \eta_{as} = 1\} \quad (4.1)$$

Para tratar as sobreposições de horário define-se, pela expressão (4.2), o conjunto  $S_2^a(X)$  das salas disponíveis para a aula  $a$ , dado as alocações já realizadas em  $X$ . O conjunto  $C^a$  é o subconjunto de aulas que possuem sobreposição de horário com a aula  $a$ .

$$S_2^a(X) \doteq \{s \in S \mid \forall a_i \in C^a \rightarrow X(a_i) \neq s\} \quad (4.2)$$

Para que uma sala  $s$  seja uma alocação factível para uma aula  $a$ , considerando as alocações  $X$  já realizadas, ela deverá ser uma sala possível e estar disponível simultaneamente. Define-se então o conjunto  $\overline{S^a}(X)$ , pela expressão (4.3), das salas factíveis para a aula  $a$ , dado o mapa atual  $X$ .

$$\overline{S^a}(X) \doteq S_1^a \cap S_2^a(X) \quad (4.3)$$

A melhor sala  $s^*(X, a)$  para ser alocada para uma aula  $a$ , considerando o mapa atual  $X$ , será a que possuir menor valor de  $\Delta_a(X, s)$ , indicada pela expressão (4.4).

$$s^*(X, a) \doteq \arg \min_{s \in \overline{S^a}(X)} \{\Delta_a(X, s)\} \quad (4.4)$$

A heurística construtiva proposta para o PAAS com cinco métricas constrói a solução aula a aula, portanto a ordem da escolha das aulas impacta no funcionamento do algoritmo. É possível que em alguma interação da heurística, o conjunto  $S^a(X)$  seja vazio, i.e., não existem salas factíveis para a aula  $a$  considerando as alocações que já foram realizadas em  $X$ . Caso  $\exists s^*(X, a)$ , o algoritmo para e uma nova ordenação aleatória para o

conjunto  $A$  é proposta. O procedimento é ilustrado pelo Algoritmo 4.2.

---

**Algoritmo 4.2:** Pseudocódigo da Heurística Construtiva

---

**Entrada:** Instância do PAAS  
**Saída:** Solução  $X$

- 1 Criar  $w \in \{0, 1\}^{|C| \times |S|}$  atribuindo valor nulo para todas as entradas;
- 2 **para cada**  $a \in A$  **faça**
- 3     Obter  $S^a(X)$  (Expressão 4.3);
- 4     Obter  $s^*(X, a)$  (Expressão 4.4);
- 5     **se**  $\nexists s_a^*$  **então**
- 6         **retorna** *Erro*
- 7     **senão**
- 8         Fazer  $X(a) = s_a^*$ ;
- 9         Obter  $t_a$  turma da aula  $a$ ;
- 10        **para cada**  $c \in C \mid \Upsilon_{t_a c} = 1$  **faça**
- 11            Fazer  $w_{cs_a^*} = 1$ ;
- 12        **fim**
- 13     **fim**
- 14 **fim**
- 15 **retorna**  $X$

---

As soluções construídas pelo Algoritmo 4.2 são mapas  $X : A \rightarrow S$  que atendem às duas propriedades da Definição 1 e portanto são soluções factíveis do PAAS. Quando há o retorno de Erro (linha 6) o algoritmo para e uma nova ordenação de  $A$  aleatória é proposta. Na linha 10,  $\Upsilon$  é a matriz binária que associa turmas e currículos, definida no Capítulo 3, dado de entrada do problema.

## 4.2 Busca Local

A busca local é um procedimento iterativo, baseado em vizinhança, de refinamento de soluções para problemas de otimização. Seja o conjunto  $\mathbb{X} = \{X : A \rightarrow S \mid X \text{ é factível}\}$  o conjunto de todas as soluções factíveis para um dado problema e  $F : \mathbb{X} \rightarrow \mathbb{R}$  uma função avaliação, que associa a cada solução  $X$  um valor de função objetivo  $F(X)$ .

Uma vizinhança será uma aplicação  $\mathfrak{N} : \mathbb{X} \rightarrow \mathbb{X}$  que leva uma solução  $X$  para uma solução vizinha  $\overline{X}$ . A busca local parte de uma solução inicial  $X^0$ , usualmente obtida por uma heurística construtiva, e obtém uma solução vizinha  $\overline{X^0}$ . Caso a solução obtida possua melhor valor de função objetivo, por exemplo,  $F(\overline{X^0}) < F(X^0)$  para problemas de minimização, ela é tomada como solução corrente.

A vizinhança é aplicada sobre a solução corrente, até que uma nova solução de melhor valor de função objetivo não possa ser encontrada. Caso o algoritmo não obtenha novas

soluções, por meio da vizinhança utilizada, a solução obtida é um ótimo local por esta vizinhança (LENSTRA, 1997). Uma ilustração deste procedimento é dado pela Figura 2.

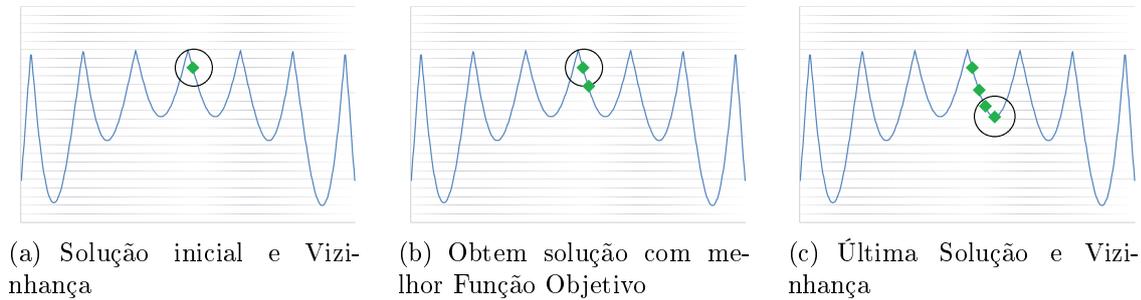


Figura 2: Exemplo do funcionamento da Busca Local

A Figura 2 exemplifica uma aplicação de busca local para um problema de minimização. A Figura 2(a) é a solução inicial, com a vizinhança ilustrada pelo círculo. A Figura 2(b) ilustra a segunda solução obtida dentro da vizinhança e a Figura 2(c) é o fim do processo, em que um ótimo local é obtido.

A forma de explorar a vizinhança, i.e., escolha da solução corrente, difere a busca local em dois tipos, busca de primeira melhora e busca de máxima melhora. A busca de primeira melhora (*first improvement*) substitui a solução corrente pela primeira solução vizinha que possua melhora na função objetivo. A busca de máxima melhora (*best improvement*) considera toda a vizinhança da solução corrente e a melhor solução da vizinhança é escolhida caso melhore o valor da função objetivo da solução corrente.

Na literatura, para tratar o PAAS, são encontrados algoritmos que utilizam busca local, usualmente integrando-a a outros métodos ou estratégias. Alguns trabalhos considerados foram: Algoritmos Meméticos ((SOUZA; SCARPIN, 2014)), Busca Tabu ((SOUZA et al., 2002a) (SUBRAMANIAN et al., 2011)) e Recozimento Simulado ((SOUZA et al., 2002a) (MARTINEZ-ALFARO; FLORES-TERAN, 1998) (NASCIMENTO et al., 2005) (KRIPKA; KRIPKA, 2010)). Nestes trabalhos existem duas estruturas de vizinhança normalmente utilizadas, denotados por movimento de troca e movimento de alocação.

O movimento de troca, ilustrado pela Figura 3, escolhe 2 aulas que possuam sobreposição de horário, alocadas em salas diferentes, e inverte as salas de ambas. Este movimento não pode gerar sobreposição de horários nas salas de origem e de destino das aulas, como observa-se na solução obtida na Figura 3(c).

A Figura 3(a) ilustra a solução inicial, na Figura 3(b) as aulas A31 (alocada na sala S2) e A29 (alocada na sala S4) são escolhidas para a troca. A Figura 3(c) ilustra o movimento de troca, em que a aula A31 é alocada na sala S4 e a aula A29 é trocada para

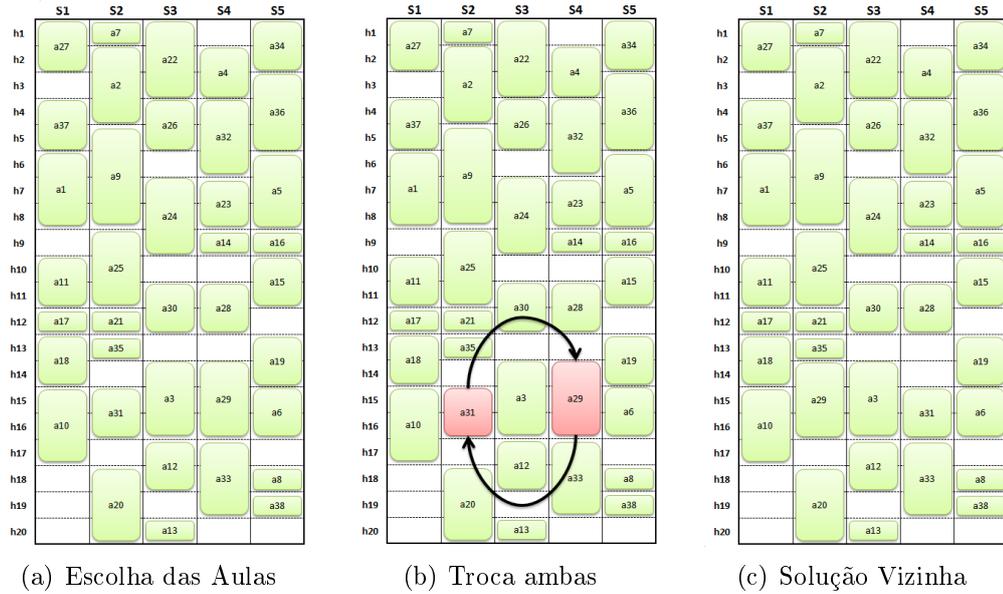


Figura 3: Exemplificação do movimento de Troca

sala S2. Este movimento só foi possível pois as aulas não geravam sobreposição de horário nas salas destino, e.g., as aulas A27 (S1) e A7 (S2) não poderiam ser escolhidas.

Outra função vizinhança comumente encontrada é o movimento de alocação. Este movimento, ilustrado na Figura 4, muda a sala em que uma aula já esta alocada. Da mesma forma que o movimento de troca, esta operação não pode gerar sobreposições de horário na sala de destino. No exemplo, a aula 20 estava alocada na sala S2 e como a sala S1 estava vaga no horário, a realocação foi possível, obtendo uma nova solução.

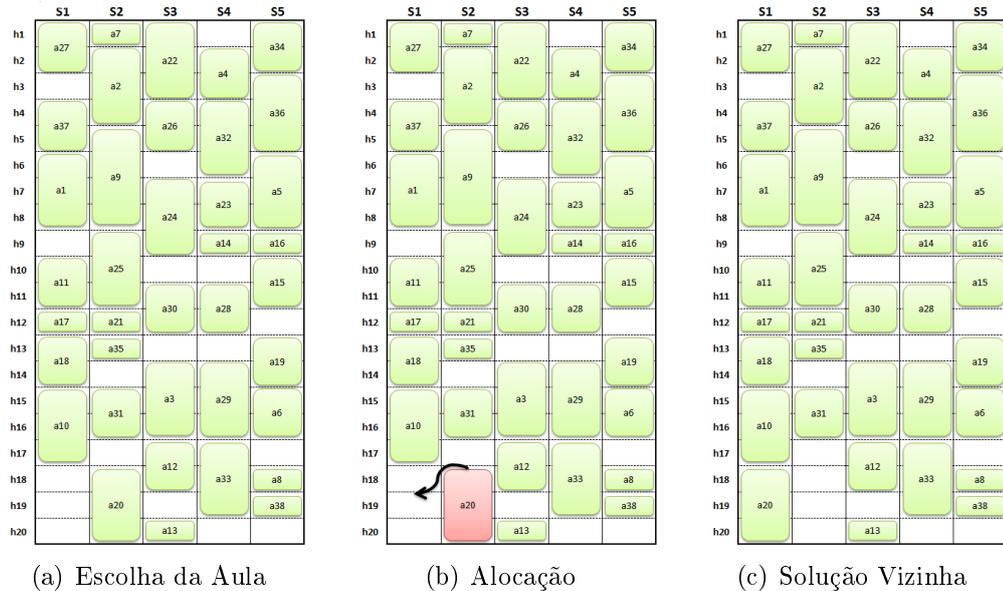


Figura 4: Exemplificação do movimento de Alocação

Para esta dissertação foi proposta uma nova estrutura de vizinhança, com uma es-

estratégia de busca de primeira melhora. Esta vizinhança parte de uma solução factível e escolhe uma aula  $a$  aleatoriamente, esta aula é movida da sala de origem  $s$  para uma sala de destino  $\hat{s}$  que possua a capacidade e os recursos para a realização da aula, i.e.,  $\hat{s} \in S^a \setminus \{s\}$ .

A vizinhança implementada é baseada na vizinhança proposta por Saviniec et al. (2013) para o problema de currículos escolares. Esta vizinhança parte de uma solução inicial  $X$  e move uma aula  $a$  para uma sala  $s \in S^a$ . Como o movimento só considera salas  $s \in S^a$  a primeira propriedade da Definição 1 é garantida.

Este movimento porém pode gerar sobreposições de horário na sala  $s$  de destino. Seja portanto  $\overline{C^a}(X) \subset C^a$  o conjunto das aulas  $\bar{a}$  que estão em sobreposição de horário com a aula  $a$  na solução  $X$ . Se  $\overline{C^a}(X) = \emptyset$ , a solução obtida com o movimento é factível, senão, para cada aula  $\bar{a} \in \overline{C^a}(X)$  a função é recursivamente chamada para mover estas aulas. O Algoritmo 4.3 ilustra o método recursivo.

---

**Algoritmo 4.3:** Função Recursiva da Vizinhança da Busca Local

---

**Entrada:** Solução  $X$ , Aula  $a$

**Saída:** Solução Vizinha  $\bar{X}$

```

1 Função MoveAula( $X, a$ ) :
2    $X' = X$ ;
3   Obter  $s \in S$ , tal que  $X'(a) = s$ ;
4   Obter  $S^a \subset S$  das salas possíveis para aula  $a$ ;
5   se  $|S^a| = 1$  então
6     retorna  $X$ 
7   senão
8     Escolher  $\bar{s} \in S^a \setminus \{s\}$ ;
9     Fazer  $X'(a) = \bar{s}$ ;
10    para cada  $a' \in C^a$  faça
11       $S^{a'} = S^{a'} - \{\bar{s}\}$ ;
12    fim
13    Obter  $\overline{C^a}(X) \subset A$  das aulas com sobreposição;
14    se  $\overline{C^a}(X) = \emptyset$  então
15      retorna  $X'$ 
16    senão
17      para cada  $\bar{a} \in \overline{C^a}(X)$  faça
18         $X' = \text{MoveAula}(X', \bar{a})$ ;
19      fim
20      retorna  $X'$ 
21    fim
22  fim
23 fim

```

---

Para evitar ciclos nestas chamadas recursivas, uma vez que uma aula  $a$  tenha sido movida da sala  $s$  para a sala  $\bar{s}$ , ela não pode ser movida novamente até o fim da recursão.

Para que a aula  $a$  não seja movida da sala  $\bar{s}$ , impede-se que qualquer aula  $a'$  com sobreposição de horário com a aula  $a$  ( $a' \in C^a$ ) seja movida para a sala  $\bar{s}$ , como ilustrado na linha 10 do Algoritmo 4.3.

Em alguma etapa da pilha de recursão pode não haver mais salas possíveis para uma aula  $a$ , i.e.,  $|S^a| = 1$ . Neste caso o processo é interrompido e retorna a solução original como solução, linha 6. O critério de parada da recursão, linha 15, é não haver mais sobreposições na solução  $X'$ .

A Busca Local implementada utiliza esta estrutura de vizinhança com uma estratégia de busca de primeira melhora. Depois de cada iteração do Algoritmo 4.3 os conjuntos  $S^a$  retornam para o estado inicial, como ilustra a linha 6 do Algoritmo 4.4. O critério de parada utilizado é o tempo de execução.

---

**Algoritmo 4.4:** Busca Local
 

---

**Entrada:** Solução  $X$ , Tempo máximo de execução  $t_{max}$   
**Saída:** Solução Vizinha  $\bar{X}$

- 1 Fazer  $t =$  tempo inicial;
- 2 Fazer  $t_f = t + t_{max}$  tempo final;
- 3 **enquanto**  $t \leq t_f$  **faça**
- 4     Escolher  $a \in A$  ;
- 5     Obter  $X' = \text{MoveAula}(X, a)$  ;
- 6     Retornar  $S^a$  para o estado inicial ;
- 7     **se**  $F(X') < F(X)$  **então**
- 8          $X = X'$ ;
- 9     **fim**
- 10    Fazer  $t =$  tempo atual;
- 11 **fim**
- 12 **retorna**  $X$

---

A linha 7 do Algoritmo 4.4 verifica se o valor de função objetivo (utilizando a função objetivo (3.8) da Seção 3) da solução  $X'$  obtida é menor que o valor de função objetivo da solução corrente  $X$ , pois o problema é de minimização. A linha 5 chama o Algoritmo 4.3.

Esta vizinhança tem a propriedade de gerar somente soluções factíveis e é computacionalmente leve. Os movimentos de alocação e de troca originais podem ocorrer dentro desta vizinhança. A Figura 5 exemplifica brevemente o processo.

A Figura 5(a) ilustra a solução inicial fornecida. Na Figura 5(b) são feitas as escolhas da aula A24 (alocada em S3) e da sala destino S4. A aula A24 é movida para a sala S4 e verifica-se que há sobreposição com aulas A23 e A14 na sala S4, Figura 5(c).

A função recursiva é chamada para cada uma destas aulas em sobreposição, como



Figura 5: Exemplificação da nova Vizinhaça proposta

ilustra a Figura 5(d), em que a aula A23 é movida para a sala S3 e a aula A14 é movida para sala S5. Este movimento faz com que a aula A14 fique em sobreposição de horário com A16, portanto a função recursiva é chamada novamente em A16, Figura 5(e), em que S1 é escolhida como sala destino para A16. Este movimento final de A16 para S1 não gera sobreposições de horário e portanto o processo para com uma solução vizinha (Figura 5(f)).

### 4.3 Busca Tabu

A Busca Tabu (BT), proposta por Glover (1986), é uma metaheurística, baseada em busca em vizinhança munida de uma estratégia para escapar de ótimos locais. Esta estratégia permite que a solução corrente  $X$  da busca seja substituída por uma solução vizinha  $\bar{X}$  de pior valor de função objetivo. O movimento inverso, i.e., o movimento pela vizinhança que leva a solução  $\bar{X}$  de volta para a solução original  $X$ , se torna um movimento tabu. Os movimentos tabu são adicionado à Lista Tabu (LT), impossibilitando a aplicação dos mesmos, i.e., evitando ciclos. Com os movimentos de piora sendo aceitos, a metaheurística pode sair de um ótimo local.

A LT porém tem alto custo computacional para manutenção e consulta. Portanto um parâmetro comumente utilizado para a BT é o *Tabu* (tamanho máximo da LT). Caso a LT atinja o tamanho *Tabu*, a solução corrente é atualizada para a melhor solução obtida até então e a lista Tabu é esvaziada.

Um movimento que pertença a LT pode ser aceito caso o critério de aspiração seja satisfeito, i.e., caso a aplicação do movimento tabu produza uma solução de melhor valor de função objetivo que a melhor solução encontrada. Quando uma solução corrente de melhor valor de função objetivo é obtida a LT é esvaziada. A Figura 6 ilustra a metaheurística.

A Figura 2(c) ilustra o fim de uma Busca Local, em que um ótimo local foi obtido. Portanto, caso não haja uma estrutura de vizinhança que permita uma solução com melhor função objetivo ser obtida, a busca local ficará presa no ótimo local. A Figura 6(a) ilustra a primeira solução de piora aceita como solução corrente.

A Figura 6(b) exemplifica a capacidade da metaheurística de continuar ascendendo em soluções de piora, até que uma solução de melhor valor de função objetivo é obtida (Figura 6(c)). A partir deste ponto a BT tem comportamento similar à busca local e desce até um novo ótimo local, Figura 6(d), de melhor qualidade que a solução inicial da Figura 6(a).

Na literatura que trata do PAAS, a BT é utilizada nos trabalhos de Souza et al. (2002a) e Subramanian et al. (2011). Em ambos foram utilizadas as vizinhanças de troca e de alocação, descritas na Seção 4.2.

Para esta dissertação a vizinhança considerada foi a mesma da Busca Local (Algoritmo 4.3). Esta vizinhança recebe uma solução  $X$  e uma aula  $a$ , realocando a aula  $a$  para a sala  $\bar{s}$ . O movimento tabu impedirá que a aula  $a$  não possa ser realocada para a sala

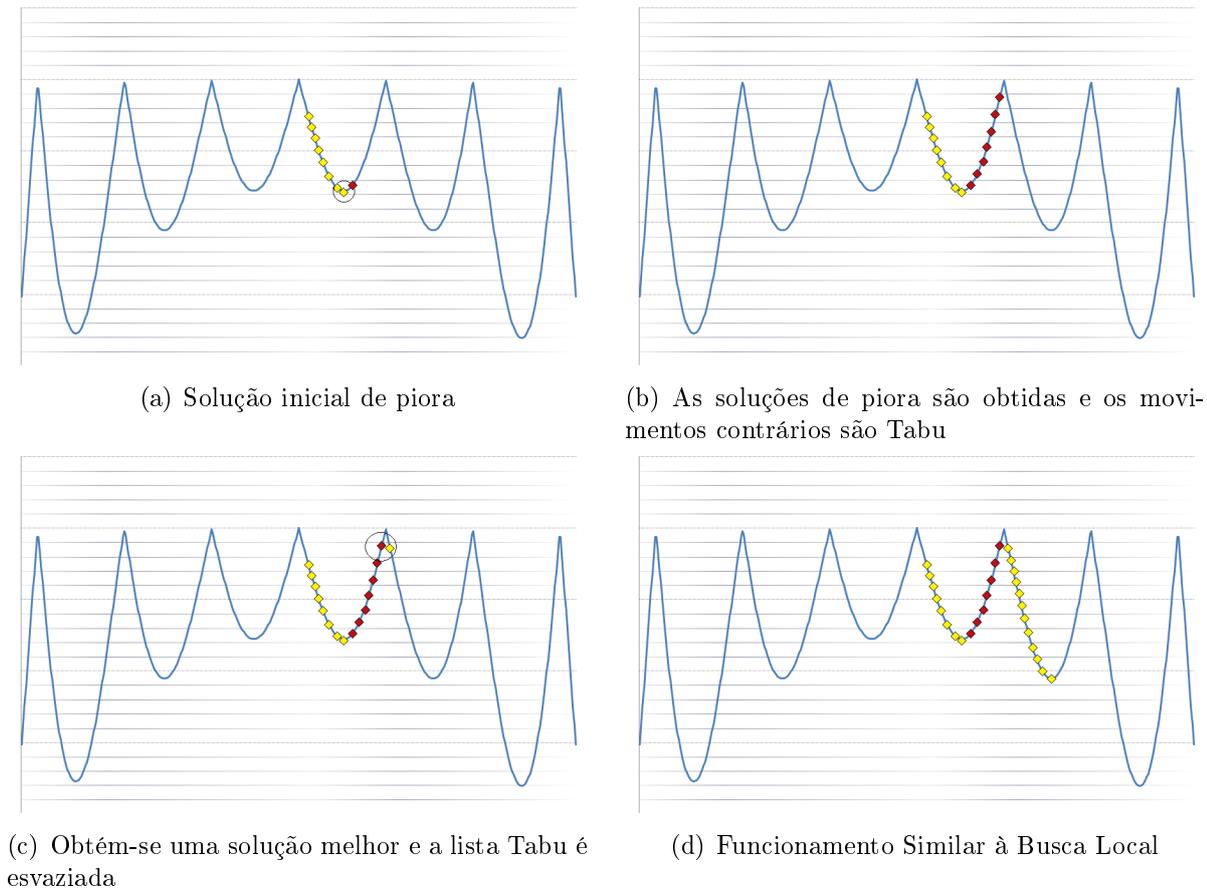


Figura 6: Exemplo do funcionamento da Busca Tabu

$s$ , enquanto este par  $(a,s)$  estiver na lista Tabu. O Algoritmo 4.5 ilustra a Busca Tabu implementada.

No Algoritmo 4.5, na linha 8 verifica-se o critério de aspiração, independente da condição tabu, caso seja atendido, atualiza a melhor solução corrente e esvazia a lista tabu. Caso contrário, verifica-se a condição tabu do movimento (linha 14). Caso o movimento seja tabu, ele é descartado e uma nova aula é amostrada, linha 7. Caso o movimento escolhido não seja tabu, verifica se há melhora na função objetivo com relação à solução corrente  $X$  (linha 15). Caso positivo o método funciona igualmente à busca local (linha 16).

Caso um movimento que não seja tabu apresente piora na função objetivo, confere se a lista tabu esta cheia (linha 18). Caso a lista esteja cheia, a solução corrente  $X$  retrocede à melhor solução obtida  $X^*$  (linha 19) e a lista tabu é esvaziada, esta foi uma decisão de projeto para esta dissertação. Caso a lista não esteja cheia, aplica-se a estratégia tabu, i.e., o movimento de piora é aceito e adicionado na lista tabu (linhas 22 e 23).

**Algoritmo 4.5:** Busca Tabu

---

**Entrada:** Solução  $X$ ,  $Tabu$ , Tempo  $t_{max}$   
**Saída:** Solução Vizinha  $\bar{X}$

- 1 Fazer  $t =$  tempo inicial;
- 2 Fazer  $t_f = t + t_{max}$  tempo final;
- 3 Fazer  $X^* = X$  melhor solução corrente;
- 4 Fazer  $LT = \emptyset$  lista tabu;
- 5 **enquanto**  $t \leq t_f$  **faça**
- 6     Escolher  $a \in A$ ;
- 7     Obter  $X' = \text{MoveAula}(X, a)$  ;
- 8     **se**  $F(X') < F(X^*)$  **então**
- 9          $X^* = X'$ ;
- 10          $X = X'$ ;
- 11          $LT = \emptyset$ ;
- 12     **senão**
- 13          $s = X'(a)$ ;
- 14         **se**  $(a, s) \notin LT$  **então**
- 15             **se**  $F(X') \leq F(X)$  **então**
- 16                  $X = X'$ ;
- 17             **senão**
- 18                 **se**  $|LT| = Tabu$  **então**
- 19                      $X = X^*$ ;
- 20                      $LT = \emptyset$ ;
- 21                 **senão**
- 22                      $X = X'$ ;
- 23                      $LT = LT + \{(a, s)\}$ ;
- 24             **fim**
- 25         **fim**
- 26     **fim**
- 27     **fim**
- 28     Fazer  $t =$  tempo atual;
- 29 **fim**
- 30 **retorna**  $X^*$

---

## 4.4 GRASP

O GRASP (*Greedy Randomized Adaptive Search Procedure*) é um procedimento iterativo para resolução de problemas de otimização, proposto por Feo e Resende (1995). O procedimento é composto de duas fases: a fase construtiva e a fase de refino. Na fase construtiva deve existir diversidade nas soluções construídas, portanto um fator aleatório deve ser inserido. Enquanto para a fase de refino o usual é a aplicação de uma busca local. A melhor solução obtida durante as fases construtiva e de refino do GRASP é armazenada e esta será a solução obtida pela metaheurística.

Para a fase construtiva os elementos são escolhidos segundo uma função míope, i.e.,

uma função que considera somente os impactos imediatos das escolhas. Estes elementos são ordenados segundo esta função e atualizados a cada iteração. É proposta uma lista restrita de candidatos *RLC* (*Restricted Candidate List*), de tamanho máximo dado pelo parâmetro  $\mathbb{L}$ , em que os elementos são ordenados segundo esta função. O fator aleatório é inserido na fase construtiva um elemento da *RLC* é escolhido ao acaso, mas não necessariamente o primeiro.

Para a fase de refino normalmente é proposta uma busca local, porém a utilização da BT no lugar da busca local para o GRASP não é incomum de se encontrar na literatura. Em Laguna e Velarde (1991), o GRASP com uma Busca Tabu foi proposto para um problema de *Just-in-Time* e Souza et al. (2004) propõe um GRASP com BT para o problema de programação de cursos universitários. Aplicações desta hibridização para outros problemas podem ser encontradas em Peiró et al. (2014) e Silva et al. (2012).

Para esta dissertação foram propostas duas implementações para o GRASP. Para ambas, a fase construtiva é baseada na Heurística Construtiva proposta na Seção 4.1, com a adição do um fator aleatório por meio da lista *RLC*. Em uma das implementações, o refino será feito por meio da busca local proposta na Seção 4.2 e a outra será a BT da Seção 4.3.

---

**Algoritmo 4.6:** Pseudocódigo da Heurística Construtiva do GRASP

---

**Entrada:**  $\mathbb{L}$  tamanho máximo da *RLC*

**Saída:** Solução  $X$

```

1 Criar  $w_{cs} \in \{0, 1\}^{|C| \times |S|}$ ;
2 Fazer  $w_{cs} = 0$ ,  $c \in C$ ,  $s \in S$ ;
3 para cada  $a \in A$  faça
4   Obter  $\overline{S^a}(X)$  (Expressão 4.3 na Seção 4.1);
5   Ordenar  $\overline{S^a}(X)$  pela métrica  $\Delta_a(X, s)$  (Algoritmo 4.1 na Seção 4.1);
6   Obter  $\overline{s} \in S_{|\mathbb{L}|}^a$ ;
7   se  $S_{|\mathbb{L}|}^a = \emptyset$  então
8     retorna Erro
9   senão
10    Fazer  $X(a) = \overline{s}$ ;
11    Obter  $t_a$  turma da aula  $a$ ;
12    para cada  $c \in C \mid \Upsilon_{t_a c} = 1$  faça
13      Fazer  $w_{c\overline{s}} = 1$ ;
14    fim
15  fim
16 fim
17 retorna  $X$ 

```

---

A heurística construtiva proposta para o GRASP adiciona aleatoriedade por meio da *RLC*. A *RLC* proposta para o PAAS será obtida pelo conjunto  $\overline{S^a}(X)$  (expressão

4.3) e a métrica induzida por  $\Delta_a(X, s)$ , descritos na Seção 4.1. A *RLC* será a sublista  $\overline{S^a}_{\mathbb{L}}(X) \subset \overline{S^a}(X)$ , que contém os  $\mathbb{L}$  primeiros elementos de  $\overline{S^a}(X)$  ordenados segundo os valores de  $\Delta_a(X, s)$  de seus elementos.

Portanto, a principal diferença entre a heurística construtiva (Algoritmo 4.2 da Seção 4.1) e a heurística construtiva do GRASP será na escolha das salas para cada aula. Na primeira a escolha é feita por meio da expressão (4.4), porém para a construtiva do PAAS qualquer sala da lista  $\overline{S^a}_{\mathbb{L}}(X)$  é escolhida com mesma probabilidade. O Algoritmo 4.6 ilustra o processo.

Na linha 6 do Algoritmo 4.6, a sala  $s_a$  é escolhida atribuindo probabilidade  $\frac{1}{|\overline{S^a}_{\mathbb{L}}|}$  de escolha para cada sala da sublista  $\overline{S^a}_{\mathbb{L}}$ . O erro na linha 8, análogo ao erro da heurística construtiva da Seção 4.1, ocorre quando não há salas disponíveis para a aula. Na heurística construtiva ela era interrompida e requeria uma nova ordenação de aulas, porém para o caso da Construtiva do GRASP uma nova ordenação de aulas é gerada aleatoriamente e o mesmo é executado novamente.

Foram implementadas duas variantes do GRASP, uma utilizando a Busca Local (*GRASP*) e outra utilizando a Busca Tabu (*GRASP<sub>T</sub>*). A Busca Local e Busca Tabu tem como critério de parada o tempo de execução, portanto deve-se estabelecer um tempo máximo de busca,  $t_b$ , em cada iteração do GRASP.

---

**Algoritmo 4.7:** Pseudocódigo do GRASP
 

---

**Entrada:**  $\mathbb{L}$  tamanho da lista *RLC*,  $N$  número de execuções da busca,  $t_{max}$  tempo de execução total

**Saída:** Solução  $X$

```

1 Fazer  $t =$  tempo inicial;
2 Fazer  $t_b = \frac{t_{max}}{N}$  tempo de busca;
3 Fazer  $t_f = t + t_{max}$  tempo final;
4 Fazer  $X^*$  melhor solução;
5 enquanto  $t \leq t_f$  faça
6   | Obter  $X$  pela Construtiva( $\mathbb{L}$ ) (Algoritmo 4.6);
7   | Obter  $X'$  pela Busca( $X, t_b$ );
8   | se  $F(X') < F(X^*)$  então
9   |   |  $X^* = X'$ ;
10  | fim
11  | Fazer  $t =$  o tempo atual;
12 fim
13 retorna  $X^*$ 

```

---

Tanto o *GRASP* quanto o *GRASP<sub>T</sub>* irão utilizar o tempo total de execução  $t_{max}$  como critério de parada. Define-se o parâmetro  $N$  para o *GRASP* e *GRASP<sub>T</sub>*, que determina quantas vezes a busca será chamada durante a execução da metaheurística. Cada chamada

da busca terá tempo idêntico, portanto  $t_b \doteq \frac{t_{max}}{N}$  será o tempo de execução destas.

O Algoritmo 4.7 apresenta o GRASP implementado para o PAAS. A diferença entre o *GRASP* e o *GRASP<sub>T</sub>* ocorre na linha 7, em que o *GRASP* utiliza a busca local e o *GRASP<sub>T</sub>* a BT. Para o *GRASP<sub>T</sub>*, deve-se definir também o parâmetro *Tabu* de tamanho máximo da Lista Tabu da Busca Tabu.

## 4.5 Algoritmo Genético

O Algoritmo Genético (AG), proposto por Holland (1975), é uma metaheurística populacional, inspirada na teoria da evolução e seleção natural de Darwin. Segundo as teorias biológicas, os seres vivos possuem um fenótipo (do grego *phainein*, evidente, e *typos*, característica) que são suas características externas. Estas características são provenientes do genótipo (do grego *genos*, origem, e *typos*, característica), que é a constituição genética deste.

O genótipo portanto irá atribuir o indivíduo com características fenotípicas. O fenótipo do indivíduo lhe permite interagir com o meio em que esta inserido, munindo-lhe com as ferramentas para a sobrevivência e a reprodução. Os indivíduos que melhor se adaptam ao meio em que estão inseridos possuem mais probabilidade de sobreviver e se reproduzir.

A reprodução pode ser assexuada ou sexuada. A reprodução assexuada ocorre quando o indivíduo consegue reproduzir-se sem a necessidade de troca de material genético. A reprodução sexuada é baseada em gametas, que possuem metade do material genético do indivíduo, que são quando recombinados geram um novo indivíduo. Para o paralelo com o AG somente a reprodução sexuada é considerada.

A recombinação dos gametas em um novo indivíduo esta sujeita à pequenas variações ou mutações que concedem aos descendentes características fenotípicas que não presentes nos progenitores. A cada geração, os indivíduos produzem descendentes com pequenas variações, cabendo ao meio selecionar as mutações que trazem benefícios à estes. Este processo que age sobre as gerações ao longo do tempo é chamado de seleção natural (FUTUYAMA, 1992).

O paralelo, traçado por Holland (1975), para os AGs trata a solução do problema como fenótipo, determinando uma codificação genética, ou genótipo, originalmente um vetor binário unidimensional (Figura 7). O meio é representado pelo problema a ser resolvido,



Figura 7: Representação codificada de uma solução no AG

portanto soluções mais bem adaptadas ao meio são aquelas que possuem melhor valor de função objetivo. Nos AGs esta função é chamada de função *fitness* e busca-se o indivíduo mais bem adaptado, i.e., a melhor solução para o problema.

Continuando com o paralelo natural, foi proposto o operador de recombinação, que toma o genótipo dos progenitores para produzir os descendentes. Este operador, utiliza um ponto da codificação genética dos progenitores e recombina parte dos genes de cada em dois novos indivíduos. Na literatura são apresentados exemplos de alguns outros operadores de recombinação, como a recombinação de dois pontos e múltiplos pontos, para mais informações ver Sastry et al. (2005).

A Figura 8 ilustra o operador de recombinação, a Figura 8(a) representa os progenitores. Na Figura 8(b), o ponto de recombinação é ilustrado e o material genético dos progenitores é dividido em dois. Finalmente a Figura 8(c) representa os dois descendentes produzidos pelo operador de recombinação.

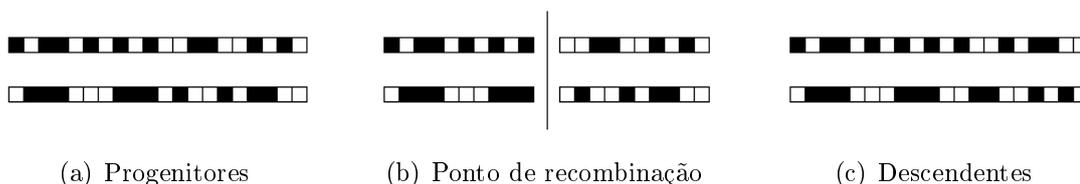


Figura 8: Operador de recombinação tradicional do AG

O operador de mutação é aplicado após o operador de recombinação, alterando o valor de um gene aleatório do indivíduo. O processo é ilustrado pela Figura 9. A Figura 9(a) representa o indivíduo que sofrerá a mutação e a Figura 9(b) indica o gene que sofrerá a mutação. O novo indivíduo obtido é ilustrado na Figura 9(c).

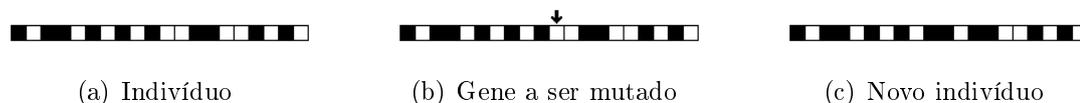


Figura 9: Operador de mutação tradicional do AG

Para cada novo indivíduo, obtido pelos processos de recombinação e mutação, é calculado o valor da função *fitness*. Os descendentes são colocados em uma população tem-

porária, juntamente com todos os indivíduos que já estavam na população. É feito então o processo de seleção natural, em que os melhores indivíduos sobrevivem e formam a população da próxima geração. O fluxograma da Figura 10 ilustra o AG.

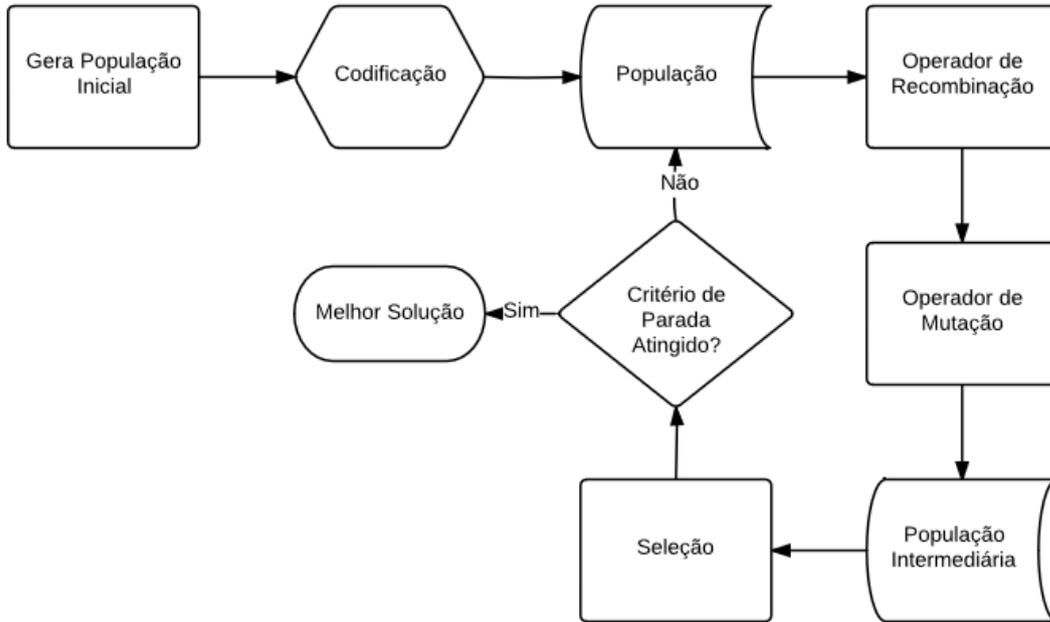


Figura 10: Fluxograma do AG

A aplicação de um AG para um problema requer:

- Escolha da codificação da solução;
- Escolha dos operadores de recombinação e mutação;
- Definição do cálculo da função *fitness*;
- Escolha dos critérios de seleção.

Para esta Dissertação, a codificação de solução utilizada foi uma matriz binária  $M \in \{0, 1\}^{|A| \times |S|}$ , em que  $A$  é o conjunto de aulas e  $S$  o conjunto de salas e  $M_{as} = 1$  se a aula  $a$  esta alocada sala  $s$ . Esta codificação é similar à  $X$  da Definição 1, caso as expressões (4.5) e (4.6) sejam satisfeitas.

$$\sum_{s \in S^a} M_{as} = 1, a \in A \quad (4.5)$$

$$\sum_{a' \in C^a} M_{as} + M_{a's} \leq 1, a \in A \text{ e } s \in S \quad (4.6)$$

Definindo  $X(a) = \arg \max_{s \in S} \{M_{as}\}$ , juntamente com a expressão (4.5), é garantida a propriedade 1 da Definição 1, i.e., cada aula  $a \in A$  possuirá uma sala  $s \in S^a$  e somente uma. A segunda expressão (4.6), considerando que todas as aulas já estejam alocadas em salas factíveis, garante que aulas em sobreposição de horário não serão alocadas na mesma sala, satisfazendo a propriedade 2 da Definição 1. Portanto uma solução  $M$  que atenda às expressões (4.5) e (4.6) será factível pela Definição 1.

A Figura 11 ilustra a codificação adotada, em que as aulas conectadas à esquerda possuem conflito de horário, e.g.,  $A01 \in C^{A03}$ . As entradas preenchidas indicam a sala alocada para as aulas, por exemplo, a aula A01 esta alocada na sala S06, bem como as aulas A05, A17 e A25 possuem sobreposição de horário entre si.

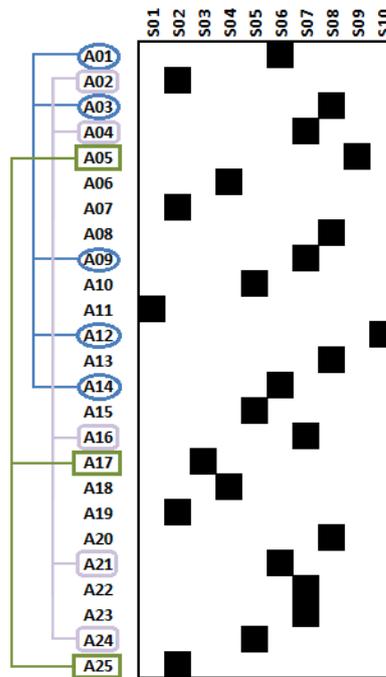


Figura 11: Codificação utilizada para o AG

O operador de recombinação proposto é uma extensão do operador de recombinação por 1 ponto tradicional para vetores binários unidimensionais. Como a representação proposta é matricial esta extensão será uma linha. Esta linha poderá ser vertical ou horizontal.

Caso a linha de recombinação seja vertical, os descendentes gerados podem ser infactíveis, mesmo tendo progenitores factíveis. Alguns descendentes podem ter duas salas alocadas para uma mesma aula e outros não terão sala alguma alocada, i.e., ambos casos violam a expressão (4.5). A recombinação por uma linha vertical e as infactibilidades que podem ocorrer são ilustradas pela Figura 12. As alocações destacadas nos descendentes

são as infecabilidades, e.g., a aula A08 alocada simultaneamente nas salas S02 e S08.

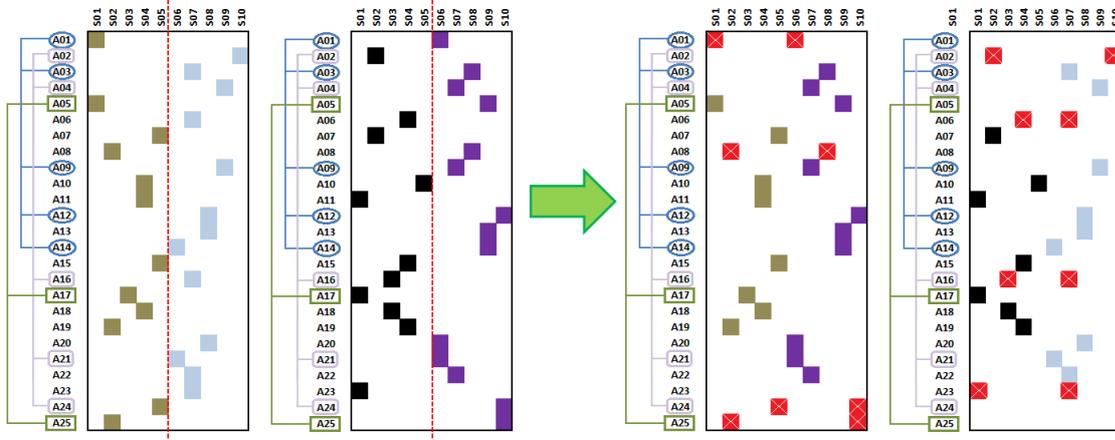


Figura 12: Recombinação utilizando uma linha vertical

Caso a linha de recombinação seja horizontal, recombinar dois indivíduos factíveis pode, também, gerar descendentes infactíveis. Neste caso, todas as aulas estarão alocadas em uma e somente uma sala. É possível, porém, que ocorram sobreposições de horário, violando a expressão (4.6). A Figura 13 ilustra a recombinação por uma linha horizontal. As entradas destacadas em vermelho indicam as infactibilidades, e.g., as aulas A09 e A14 com sobreposição de horário ambas alocadas na sala S09.

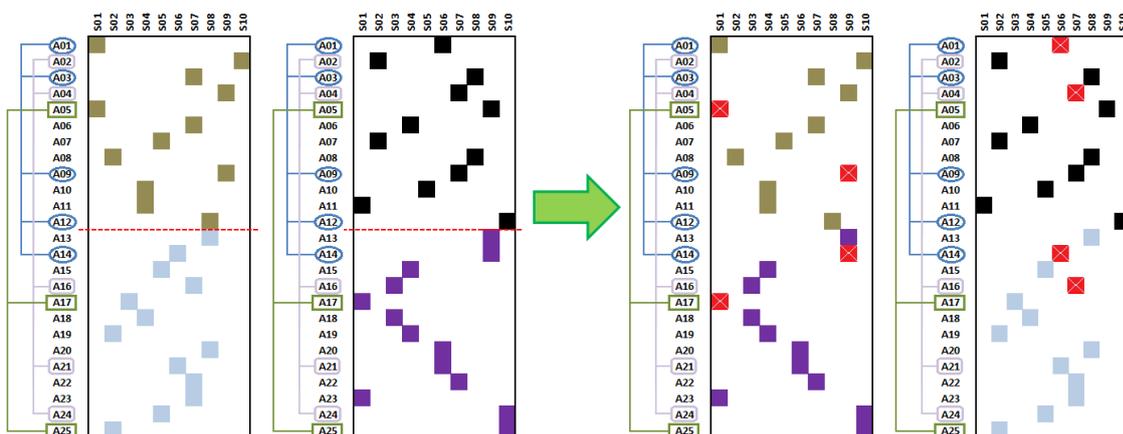


Figura 13: Recombinação utilizando uma linha horizontal

Qualquer que seja a escolha da linha de recombinação, existe a possibilidade de que sejam obtidos descendentes infactíveis, mesmo com progenitores factíveis. Em Santos et al. (2010b) são abordadas as ações a serem tomadas em caso de infactibilidade do operador de recombinação, ou seja, deve-se optar entre: descartá-las, submetê-las a um processo de factibilização ou penalizá-las na função *fitness*.

Para esta Dissertação, dado o grande número de soluções infatíveis obtidas pelo processo de recombinação, foi proposto um processo de factibilização. A recombinação utilizada foi a linha horizontal (Figura 13), partindo a solução no ponto médio e não de um ponto aleatório, por decisão de projeto. As infactibilidades obtidas serão, portanto, somente sobreposição de horário.

Considerando que ambos os progenitores são factíveis, as infactibilidades geradas serão pares de aulas  $(a_i, a_j)$  alocadas na mesma sala  $s \in S$ , tal que  $a_i \in C^{a_j}$ . O processo de factibilização seleciona aleatoriamente uma das aulas do par, por exemplo  $a_i$  e a muda da sala  $s$  para sala  $\bar{s} \in S^{a_i}$ . O processo é repetido até que não hajam mais sobreposições de horário.

A Figura 14 ilustra o processo. Na Figura 14(a), uma solução inicial é exibida com as infactibilidades (A01,A14). Na Figura 14(b), a Aula A14 é escolhida para passar pelo processo, A14 é removida da sala S06 e as suas salas possíveis estão destacadas, i.e., S04, S05, S06 e S09. Finalmente a sala sala S04 é escolhida para a aula A14, como ilustra a Figura 14(c), obtendo uma solução factível.

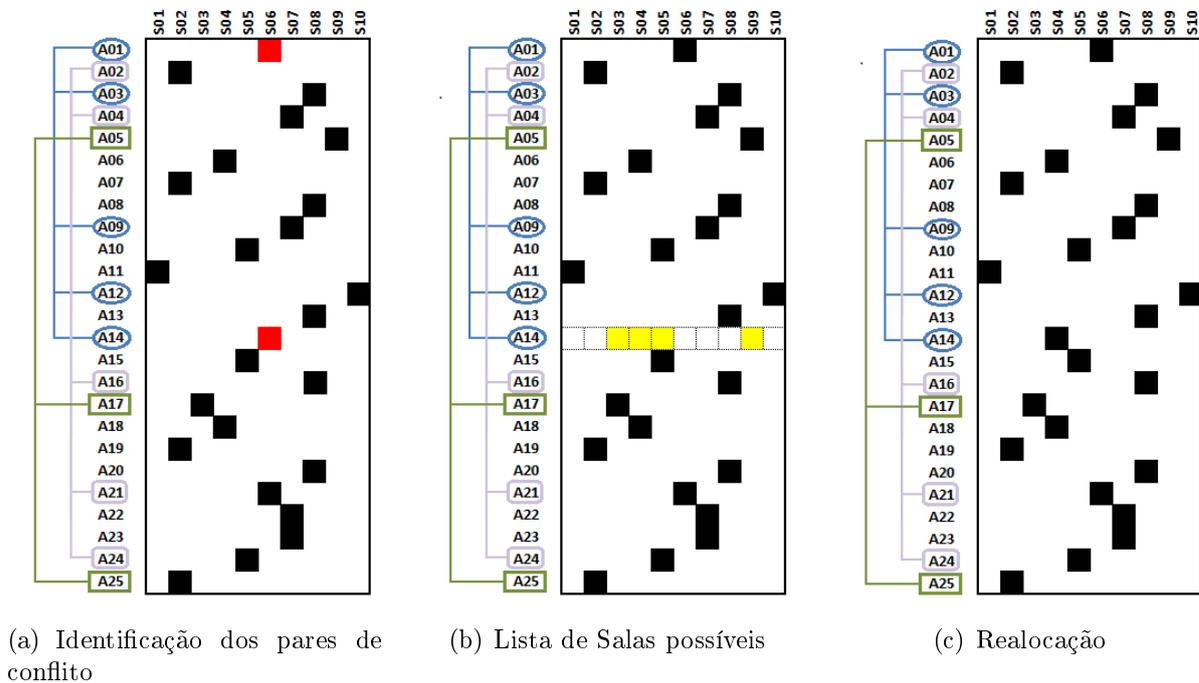


Figura 14: Processo de factibilização

O operador de recombinação recebe duas soluções factíveis e gera dois filhos. Caso os filhos não sejam factíveis, deve-se aplicar o processo de factibilização. A notação para o operador de recombinação será  $(i', j') = Recombinacao(i, j)$ .

O operador de mutação utilizado escolhe uma aula  $a$  aleatoriamente e altera a sala

alocada  $s$  para uma sala destino  $\bar{s}$ . Esta modificação deverá garantir que a sala destino  $\bar{s}$  possua capacidade e recurso requeridos pela aula  $a$ , bem como não gere sobreposição de horário quando alocada. A notação para o operador de mutação será  $i' = Mutacao(i)$ .

O AG implementado possui tamanho fixo de população, definido pelo parâmetro  $Pop$ . Após a aplicação dos operadores de recombinação, será formada uma população intermediária com mais do que  $Pop$  elementos. Ordena-se todas as soluções da população intermediária, segundo a função *fitness*, e os  $Pop$  primeiros indivíduos são selecionados para a próxima geração.

Os parâmetros do AG proposto são: o tamanho da população  $Pop$ , a taxa de recombinação  $T_r$ , a taxa de mutação  $T_m$  e o tempo máximo de execução  $t_{max}$ . A taxa de recombinação  $T_r$  é a probabilidade de que um indivíduo da população passe pelo operador de recombinação. A taxa de mutação  $T_m$  é a probabilidade de que o indivíduo gerado na recombinação passe pelo operador de mutação.

O Algoritmo 4.8 é o pseudo-código do AG implementado. Para a geração da População Inicial  $P^1$ , linha 5, foi utilizada a heurística construtiva do GRASP (Algoritmo 4.6) com parâmetro  $\mathbb{L} = 2$ , obtendo  $Pop$  soluções factíveis. Na linha 22 a população é ordenada de forma crescente pois o problema é de minimização e na linha 23 os  $Pop$  primeiros elementos são selecionados. O retorno, linha 27, do algoritmo será a primeira solução da população na última geração obtida.

## 4.6 Algoritmo Genético Compacto

O Algoritmo Genético Compacto (AGC) é um Algoritmo de Estimação de Distribuição (AED), proposto por Harik et al. (1999). Os AEDs são uma classe de algoritmos inspirados nos AGs em que os operadores genéticos são substituídos por técnicas de estimação. Especificamente no AGC a população é representada por um modelo probabilístico.

O AGC utiliza a mesma codificação binária para representar as soluções apresentada para o AG na Seção 4.5, ou seja, uma matriz binária  $M \in \{0, 1\}^{|A| \times |S|}$ , em que  $A$  é o conjunto de aulas e  $S$  o conjunto de salas. Analogamente ao caso do AG, as expressões (4.5) e (4.6) da Seção 4.5 devem ser atendidas, para que a solução representada por  $M$  seja factível. A matriz  $M$  será denotada por solução doravante.

O modelo probabilístico, que substitui a população no AGC, será denotado por  $P \in [0, 1]^{|A| \times |S|}$ . A probabilidade expressa por  $P_{as}$  é interpretada como a probabilidade de

**Algoritmo 4.8:** Pseudocódigo do Algoritmo Genético.

---

**Entrada:**  $Pop$  tamanho da população,  $T_r$  taxa de recombinação,  $T_m$  taxa de mutação,  $t_{max}$  tempo máximo de execução

**Saída:** Solução  $M$

- 1 Fazer  $t =$  tempo inicial;
- 2 Fazer  $t_b = \frac{t_{max}}{N}$  tempo de busca;
- 3 Fazer  $t_f = t + t_{max}$  tempo final;
- 4 Geração  $g = 1$ ;
- 5 Gerar população inicial  $P^g$ ;
- 6 **enquanto**  $t < t_f$  **faça**
- 7     Criar População Temporária  $P'$ ;
- 8     **para cada**  $i \in P^g$  **faça**
- 9         **se**  $Rand(seed_1) \leq T_r$  **então**
- 10             Selecionar  $j \in P^g$ , tal que  $j \neq i$ ;
- 11              $(i', j') = Recombinacao(i, j)$ ;
- 12             **se**  $Rand(seed_2) \leq T_m$  **então**
- 13                  $i' = Mutacao(i')$ ;
- 14             **fim**
- 15             **se**  $Rand(seed_3) \leq T_m$  **então**
- 16                  $j' = Mutacao(j')$ ;
- 17             **fim**
- 18             Adicionar  $i'$  e  $j'$  em  $P'$ ;
- 19     **fim**
- 20     **fim**
- 21     Adicionar  $P'$  em  $P^g$ ;
- 22     Ordenar  $P^g$  de forma crescente pela função  $fitness$ ;
- 23     Truncar  $P^g$  para tamanho  $Pop$ ;
- 24      $g = g+1$ ;
- 25     Fazer  $t$  o tempo atual;
- 26 **fim**
- 27 **retorna**  $P_1^g$

---

alocar a aula  $a$  na sala  $s$ . Pela expressão (4.5), somente uma das salas poderá ser escolhida para cada aula, portanto define-se  $\sum_{s \in S} P_{as} = 1, \forall a \in A$ .

A iniciação do modelo probabilístico é feita de forma que todas as salas possuam a mesma probabilidade de escolha. Considera-se, porém, somente as salas  $s$  que tem capacidade e recurso para atender a aula  $a$ , i.e.  $s \in S^a$ . O modelo probabilístico inicial  $P^0$  será dado pela expressão (4.7).

$$P_{as}^0 = \begin{cases} \frac{1}{|S^a|}, & \forall s \in S^a \\ 0, & \forall s \notin S^a \end{cases} \quad (4.7)$$

As soluções obtidas no AGC são amostradas do modelo probabilístico, espera-se que sejam factíveis. A iniciação do modelo probabilístico, expressão (4.7), tem por objetivo

garantir que a expressão (4.5) seja verdadeira. A expressão (4.6) será abordada na técnica de amostragem, a fim de garantir soluções factíveis.

A técnica de amostragem obtém uma solução  $M$ , a partir do modelo probabilístico  $P$ . A solução  $M$  deverá ser iniciada por meio da expressão (4.8), fazendo  $M_{as} = 0$  para todo  $s \notin S^a$ . Esta iniciação será conveniente para tratar as sobreposições de horário posteriormente.

$$M_{as} = 0, a \in A, s \notin S^a \quad (4.8)$$

A amostragem será inspirada na heurística construtiva da Seção 4.1, portanto em cada iteração uma aula  $a \in A$  é alocada em alguma sala  $s \in S^a$ . Porém no AGC a escolha da sala  $s$  para a aula  $a$  depende da probabilidade  $P_{as}$ . A Figura 15 ilustra o modelo probabilístico, com as aulas no eixo horizontal e uma probabilidade associada para cada sala, demonstrada pelas cores, e.g., a sala S10 possui aproximadamente 40% de chance de ser escolhida para a aula A01.

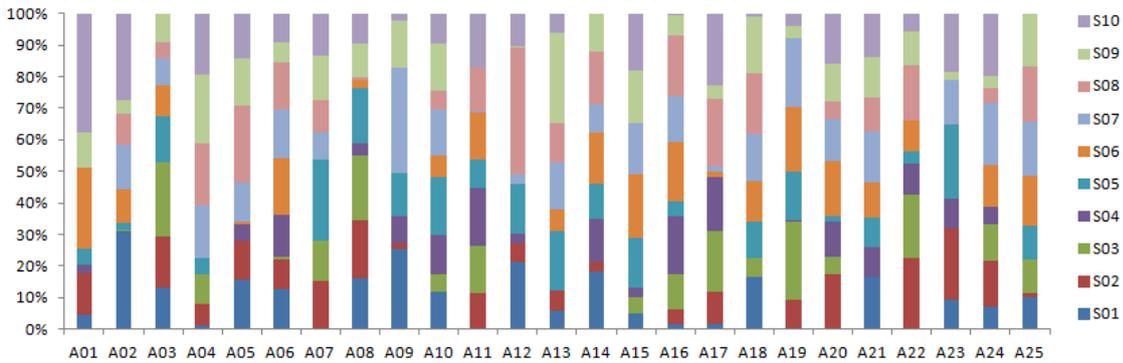


Figura 15: Representação do modelo probabilístico  $P$

Visando garantir a expressão (4.6), uma vez escolhida a sala  $s$  para a aula  $a$ , as aulas  $a' \in C^a$  devem ser proibidas de serem alocadas em  $s$ . Portanto para cada aula  $a'$  com sobreposição de horário com a aula  $a$ , faz-se  $M_{a's} = 0$ . Esta estratégia, bem como nos casos da heurística construtiva da Seção 4.1 e da construtiva do GRASP da Seção 4.4, depende da ordem com que as aulas  $a$  são alocadas.

Portanto para o AGC foram propostas duas formas de ordenação do conjunto de aulas  $A$ . A primeira ordenação é uma ordenação pelo número de salas possíveis, i.e., crescente com relação à  $|S^a|$ . Assim aulas que possuem menos opções de sala são escolhidas primeiro. A outra ordenação será completamente aleatória. A escolha da ordenação a ser aplicada será dada pelo parâmetro binário  $Ordem(A)$ , em que  $Ordem(A) = 1$  indica a ordenação

por opção de sala e  $Ordem(A) = 0$  a ordenação aleatória.

A Figura 16 ilustra o processo. A solução  $M$  é iniciada segundo a expressão (4.8), representada na Figura 16(a). As salas vazias são representadas pelas entradas -1 e as salas indisponíveis são marcadas por 0, e.g., a sala S03 esta indisponível para a aula A01. As informações de sobreposição de horário estão à esquerda nesta figura, e.g., as aulas A01 e A03 possuem sobreposição de horário.

Na Figura 16(b) a ordenação por opção de sala é exibida, ou seja, as aulas com menos salas disponíveis são escolhidas primeiro. Nesta figura são exibidas as probabilidades de escolha de cada sala para a aula A01. A sala S09 é sorteada para a aula A01, como indica a estrela.

A Figura 16(c) ilustra o processo de alocação, em que a aula A01 é alocada na sala S09, i.e.,  $M_{A01,S09} = 1$ . Faz-se  $M_{A01,s'} = 0$  com  $s' \in S \setminus \{S09\}$ . Como as aulas A03, A09, A12 e A14 possuem sobreposição de horário com a aula A01, faz-se  $M_{A03,S09} = M_{A09,S09} = M_{A12,S09} = M_{A14,S09} = 0$ . O processo continua a partir da aula A09 até que todas as aulas sejam alocadas.

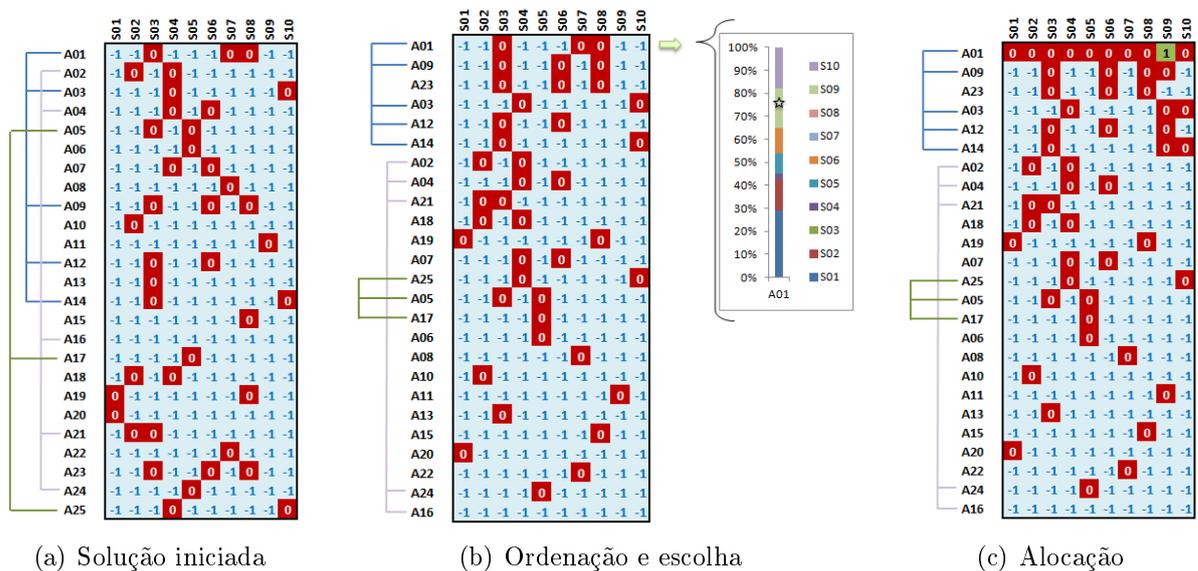


Figura 16: Processo de amostragem de solução para o AGC

Pode-se, porém, chegar à uma aula  $a \in A$  em que todas as salas possíveis já estão ocupadas e portanto, uma solução factível não poderá ser amostrada. Neste caso, para que o método de amostragem não seja encerrado, a aula  $a$  é alocada em uma sala  $s \in S^a$ , escolhida aleatoriamente. Desta forma somente a expressão (4.6) é violada, porém não há violação da expressão (4.5).

Portanto o processo de amostragem obtém soluções *quasi*-factíveis, pois eventualmente uma solução infactível é amostrada. Para uma aula  $a \in A$  é possível identificar as salas indisponíveis. Seja  $\overline{S^a}(M) \subset S$  o conjunto das salas que estão disponíveis para a aula  $a$  na solução  $M$ , dado pela expressão (4.9). As salas indisponíveis são dadas por  $S \setminus \overline{S^a}(M)$ .

$$\overline{S^a}(M) = \{s \in S \mid M_{as} \neq 0\} \quad (4.9)$$

As salas disponíveis devem possuir probabilidade não nula de serem amostradas, caso contrário uma solução factível não poderá ser amostrada. Portanto, para uma aula  $a$ , caso  $\sum_{s \in \overline{S^a}(M)} P_{as} = 0$ , aloca-se  $a$  em uma sala  $s \in S^a$ , escolhida aleatoriamente. O Algoritmo 4.9 ilustra o processo de amostragem no modelo probabilístico  $P$ , com a ordenação  $Ordem(A)$ .

---

**Algoritmo 4.9:** Pseudocódigo para obtenção de soluções (*Quasi*-)Factíveis.

---

**Entrada:** Modelo probabilístico  $P$ ,  $Ordem(A)$  ordenação de  $A$

**Saída:** Solução  $M$

- 1 Ordenar aulas  $A$  segundo  $Ordem(A)$ ;
  - 2 Iniciar solução  $M$  usando a expressão (4.8);
  - 3 **para cada**  $a \in A$  **faça**
  - 4     Faça a variável temporária  $\lambda = \sum_{s \in \overline{S^a}(M)} P_{as}$ ;
  - 5     **se**  $\lambda > 0$  **então**
  - 6         | Escolha  $\bar{s}$  com viés de probabilidade dado por  $P_{as}$ ;
  - 7     **senão**
  - 8         | Escolha  $\bar{s}$  qualquer tal que  $s' \in S^a$ ;
  - 9     **fim**
  - 10     Faça  $M_{a\bar{s}} = 1$ ;
  - 11     Faça  $M_{as} = 0, \forall s \in S$  e  $s \neq \bar{s}$ ;
  - 12     Faça  $M_{a's} = 0, \forall a' \in C^a$ ;
  - 13 **fim**
  - 14 **retorna**  $M$
- 

Na linha 2 do Algoritmo 4.9, a iniciação impede que aulas sejam alocadas em salas sem capacidade ou recurso. A linha 4 define a variável  $\lambda$  que é a soma de todas as probabilidades de  $P$  para os pares  $(a,s)$  que já foram proibidos, i.e., caso  $\lambda = 0$  todas as salas que ainda possuem probabilidade de ser escolhidas já foram ocupadas. A linha 6 utiliza o modelo probabilístico para escolha da sala, enquanto a linha 8 escolhe uma sala possível, ignorando as sobreposições de horário.

O processo é *Quasi*-Factível, pois ainda é possível amostrar uma solução infactível. Para o AGC foi adotado uma penalização na função *fitness* para soluções infactíveis. A

função *fitness* para o AGC será o valor da função objetivo do modelo  $F(M)$  (expressão (3.8)) acrescida de uma penalização  $\mathbb{M}$ , como indica a expressão (4.10).

$$fitness(M) = \begin{cases} F(M), & \text{Se } M \text{ for factível} \\ F(M) + \mathbb{M}, & C.C. \end{cases} \quad (4.10)$$

O modelo probabilístico  $P$  será atualizado a cada iteração e optou-se por um torneio dois-a-dois. Duas soluções são amostradas de  $P$  utilizando o Algoritmo 4.9 e calcula-se o *fitness* pela expressão (4.10). A solução com menor *fitness* será a vencedora do torneio, denotada por  $M^v$ , e a outra solução será a perdedora, denotada por  $M^p$ .

A atualização do modelo probabilístico de uma geração  $i$  para a próxima irá utilizar o parâmetro  $Pop$  de tamanho da população virtual, bem como o resultado do torneio. O modelo irá adicionar viés estatístico na direção da solução vencedora  $M^v$ . A atualização é feita segundo a expressão (4.11).

$$P_{as}^{i+1} = \begin{cases} \min\{1, P_{as}^i + \frac{1}{Pop}\} & , \text{ se } M_{as}^v = 1 \text{ e } M_{as}^p = 0 \\ \max\{0, P_{as}^i - \frac{1}{Pop}\} & , \text{ se } M_{as}^v = 0 \text{ e } M_{as}^p = 1 \\ P_{as}^i & \text{c.c.} \end{cases} \quad (4.11)$$

No AGC originalmente proposto por Harik et al. (1999) é previsto que, utilizando uma técnica de atualização similar à dada pela expressão (4.11), o modelo probabilístico  $P$  convergirá para uma solução do problema. Porém como existe uma modificação determinística que independe da probabilidade original, nas linhas 11 e 12 do Algoritmo 4.9, o modelo probabilístico pode não convergir para uma solução.

Define-se, pela expressão (4.12), uma função de dispersão para o modelo probabilístico  $P$ . Esta dispersão é tal que  $\xi(M) = 0$  para qualquer solução factível e para qualquer iteração  $\xi(P^i) \geq \xi(P^{i+1})$ . Portanto o modelo probabilístico terá convergido quando  $\xi(P) = 0$ .

$$\xi(P) = 1 - \frac{\sum_{a \in A} \sum_{s \in S} P_{as}^2}{|A|} \quad (4.12)$$

O Algoritmo 4.10 ilustra o AGC proposto. O critério de parada, linha 6, será a convergência do modelo probabilístico ( $\xi(P) = 0$ ) ou o tempo máximo de execução  $t_{max}$ , o que o correr primeiro. A iniciação, linha 4, utiliza a expressão (4.7). O critério de parada, linha 6, utiliza tanto o tempo de execução como o critério de dispersão. A atualização, linha 19, depende do resultado do torneio, linha 17, e atualiza o modelo pela expressão (4.11).

---

**Algoritmo 4.10:** Pseudocódigo do Algoritmo Genético Compacto.
 

---

**Entrada:**  $Pop$  Tamanho da população virtual,  $Ordem(A)$  Ordenação de  $A$ ,  $t_{max}$  Tempo total de execução

**Saída:** Solução  $M$

```

1 Fazer  $t =$  tempo inicial;
2 Fazer  $t_f = t + t_{max}$  tempo final;
3 Fazer  $M$  melhor solução;
4 Gerar modelo probabilístico inicial  $P^0$  pela Expressão (4.7);
5  $i=0$ ;
6 enquanto  $t < t_f$  faça
7   se  $\xi(P^i) = 0$  (Expressão (4.12)) então
8     retorna  $M$ ;
9   fim
10  Amostrar  $M^1$  e  $M^2$  pelo Algoritmo 4.9;
11  se  $fitness(M^1) < fitness(M)$  então
12     $M = M^1$ ;
13  fim
14  se  $fitness(M^2) < fitness(M)$  então
15     $M = M^2$ ;
16  fim
17  Realizar torneio com  $M^1$  e  $M^2$ ;
18   $i = i + 1$ ;
19  Atualizar o modelo probabilístico  $P^i$  pela Expressão (4.11);
20  Fazer  $t =$  tempo atual;
21 fim
22 retorna  $M$ ;
```

---

---



---

## CAPÍTULO 5

---

### ABORDAGEM MULTI-OBJETIVO

A resolução de problemas reais de otimização envolvem, usualmente, vários objetivos que, por vezes, são conflitantes. Outro grande problema que usualmente emerge com relação à estes objetivos é a incomensurabilidade, i.e., a unidade de medida de cada um difere. Para a abordagem mono-objetivo, vista no Capítulo 4, estas questões são solucionadas por meio de uma função ponderada dos  $d$  objetivos  $F(X) = \sum_{i=1}^d w_i f_i(X)$ , em que  $f_i(X)$  é o valor para cada objetivo da solução  $X$  e  $w_i$  é um peso associado ao mesmo.

A utilização da função ponderada institui uma ordem total no espaço de soluções factíveis  $\mathbb{X}$ , i.e.,  $\forall X_i, X_j \in \mathbb{X} \Rightarrow F(X_i) < F(X_j)$  ou  $F(X_i) = F(X_j)$  ou  $F(X_i) > F(X_j)$ . Esta ordem dada por  $F$  permite, para um problema de minimização (maximização), buscar-se uma solução  $X^* = \arg \min_{X \in \mathbb{X}} \{F(X)\}$  ( $X^* = \arg \max_{X \in \mathbb{X}} \{F(X)\}$ ). Porém depende intrinsecamente da escolha dos pesos  $\mathbf{w} = (w_1, \dots, w_d)$ .

As soluções  $X$  podem ser expressas como um ponto no espaço de objetivos  $Z$ . Este espaço de objetivos é um subespaço de  $\mathbb{R}^d$ , com  $d$  o número de objetivos, em que se associa a cada objetivo um eixo. Assim,  $\mathbf{z}(X) = (f_1(X), \dots, f_d(X))$  é o ponto em  $Z$  associado à solução  $X$  e independe da escolha dos pesos  $\mathbf{w}$ . A Figura 17 ilustra a representação de uma solução  $X$  em  $Z$ , para um problema com dois objetivos.

Uma solução  $X$  irá ter valor de  $F(X)$  distinto, dependendo da escolha de  $\mathbf{w}$ . Para ilustrar os impactos da escolha dos pesos  $\mathbf{w}$ , tome, por exemplo, um problema de mini-

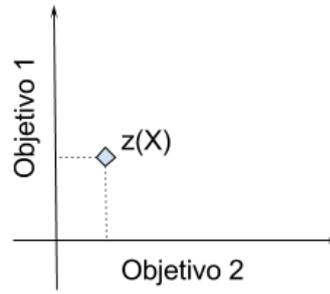


Figura 17: Representação da solução  $X$  no espaço de objetivos  $Z$

mização com dois objetivos  $d = 2$ . Assumindo  $\mathbf{w}$  unitário, define-se  $\mathbf{w} = (\sin(\theta), \cos(\theta))$ , para  $0 \leq \theta < \frac{\pi}{2}$ .

Portanto  $F(X) = f_1(X) \cos(\theta) + f_2(X) \sin(\theta)$ . Tome duas soluções  $X_1, X_2 \in \mathbb{X}$ , com  $X_1 \neq X_2$ , definidas por  $z(X_1) = \{1, 4\}$  e  $z(X_2) = \{2, 4 - \frac{\sqrt{3}}{3}\}$ . Caso  $\theta = \frac{\pi}{6}$ , tem-se  $F(X_1) = F(X_2)$  e seriam ambas soluções ótimas para a abordagem mono-objetivo. Caso  $\theta = \frac{5\pi}{12}$ , temos  $F(X_1) > F(X_2)$ . Finalmente, caso  $\theta = \frac{\pi}{3}$ , temos  $F(X_1) < F(X_2)$ . Estes casos são ilustrados pela Figura 18.

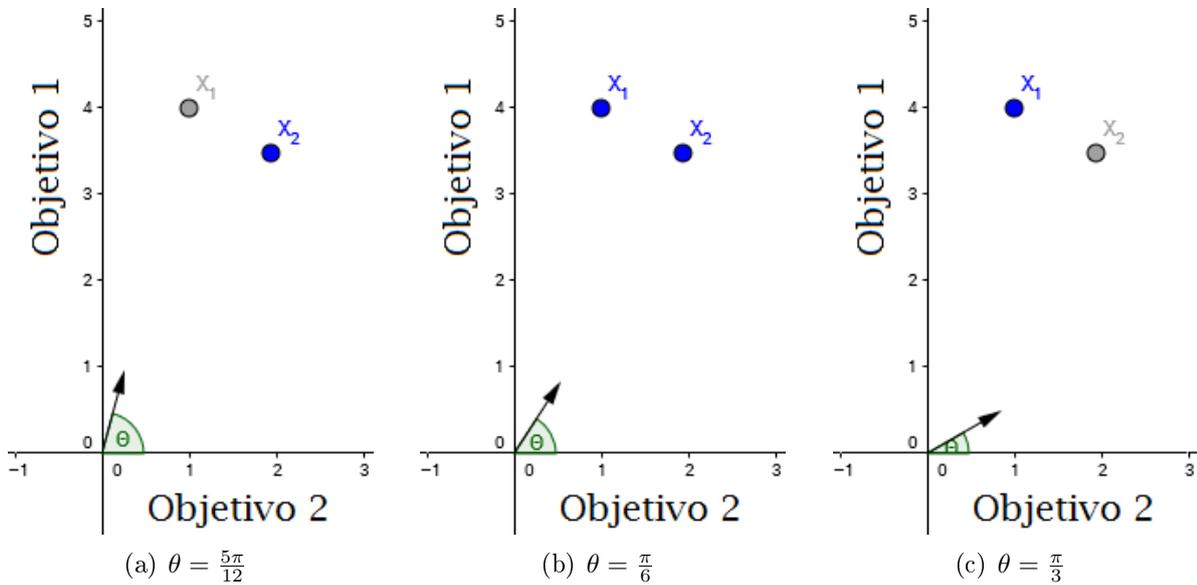


Figura 18: Ilustração do impacto da escolha dos pesos para a abordagem mono-objetivo

Na abordagem mono-objetivo pode-se desconsiderar, ou tratar como sub-ótimas, soluções que, sob outra escolha de pesos, seriam ótimas. Esta abordagem requer que o decisor, ou o proponente da solução, detenha um bom conhecimento sobre o problema tratado e as nuances em seus objetivos. A abordagem multi-objetivo tenta contornar esta questão, impondo uma ordem parcial no espaço de objetivos  $Z$ , que independe da escolha dos pesos.

Esta ordem parcial será dada pela relação de dominância  $\prec$ , como dado pela Definição 2. Na literatura esta relação é chamada *Pareto dominance* (COELLO et al., 2002). A notação desta relação é dada por  $X_1 \prec X_2$ , diz-se que  $X_1$  domina  $X_2$ .

**Definição 2.** *Seja um problema de minimização com  $d$  objetivos  $f_i, i = 1, \dots, d$ . Sejam  $Z$  o espaço de objetivos e  $\mathbb{X}$  o espaço de soluções factíveis. Para duas soluções  $X_1, X_2 \in \mathbb{X}$  define-se a **relação de dominância**  $\prec$  por:  $X_1 \prec X_2 \Leftrightarrow f_i(X_1) \leq f_i(X_2)$  e  $\exists j \in \{1, \dots, d\}$  tal que  $f_j(X_1) < f_j(X_2)$*

A Definição 2 trata de problemas de minimização, porém não há perda de generalidade. Caso o vetor de pesos possua somente entradas positivas, obtém-se  $X_1 \prec X_2 \rightarrow F(X_1) < F(X_2)$  para qualquer peso escolhido. Pois, no pior caso  $f_i(X_1) = f_i(X_2)$  para todos os objetivos e  $f_j(X_1) < f_j(X_2)$ , portanto  $F(X_1) < F(X_2)$  já que  $w_i > 0, i = 1, \dots, d$ . Outras relações de dominância e suas propriedades são apresentadas em Zitzler et al. (2003).

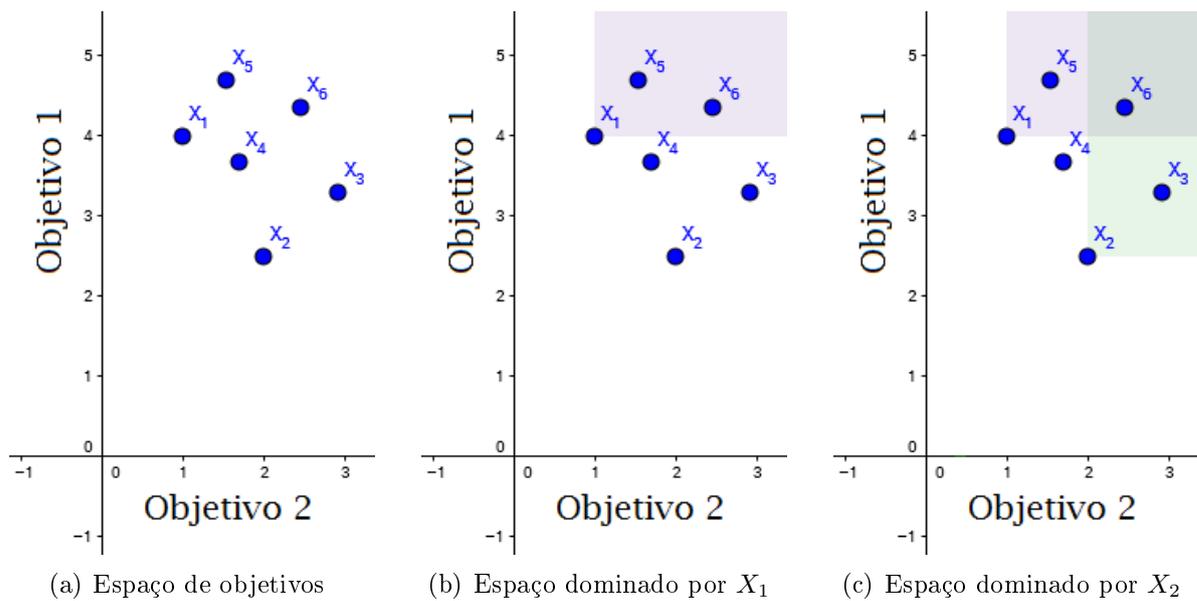


Figura 19: Relação de Dominância

A Figura 19 ilustra a ordem parcial determinada pela dominância. Na Figura 19(a) são exibidas 6 soluções,  $X_1, \dots, X_6$ . A Figura 19(b) ilustra o sub-espaço de  $Z$  dominado por  $X_1$ , e.g.,  $X_1 \prec X_5$  e  $X_1 \prec X_6$ . A Figura 19(c) ilustra o sub-espaço de  $Z$  dominado por  $X_2$ , e.g.,  $X_2 \prec X_3$  e  $X_2 \prec X_6$ . Nota-se ainda que a solução  $X_4$  não é dominada por  $X_1$  e nem por  $X_2$ , i.e.,  $X_1 \not\prec X_4$  e  $X_2 \not\prec X_4$ .

Duas soluções  $X_1$  e  $X_2$ , tais que  $X_1 \not\prec X_2$  e  $X_2 \not\prec X_1$  são ditas não-dominadas. As duas soluções não-dominadas  $X_1$  e  $X_2$  possuem um compromisso de objetivos, i.e.,  $X_1$

será melhor que  $X_2$  em algum objetivo e pior em outro. A abordagem multi-objetivo, portanto, visa obter um conjunto de soluções não-dominadas.

Define-se  $\wp(\mathbb{Y})$  a Fronteira de Pareto (FP), de um conjunto de soluções  $\mathbb{Y} \subseteq \mathbb{X}$ , como o subconjunto de soluções não dominadas, i.e.,  $\wp(\mathbb{Y}) = \{Y \in \mathbb{Y} \mid Y_i \in \mathbb{Y} \setminus \{Y\}, Y_i \not\prec Y\}$  (COELLO et al., 2002). A Figura 20 ilustra a  $\wp(\{X_1, \dots, X_6\})$ , em que as soluções  $X_1$ ,  $X_2$  e  $X_4$  constituem a FP.

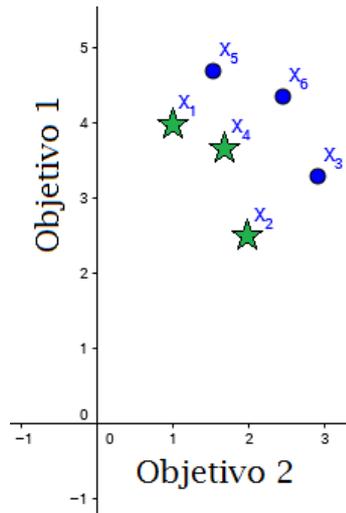


Figura 20: Exemplo da Fronteira de Pareto

A abordagem multi-objetivo tem por meta obter a Fronteira de Pareto Ótima (FPO) de um problema. A FPO é dada por  $\wp^* \doteq \wp(\mathbb{X})$ , i.e., a FP do conjunto de todas as soluções factíveis  $\mathbb{X}$ . Cabe ao decisor a escolha final, conhecendo as concessões que são feitas em uma solução em detrimento das outras.

A avaliação de desempenho dos métodos multi-objetivo é, por si só, um problema abordado na literatura, e.g., os trabalhos de Fonseca et al. (2001) e Zitzler et al. (2003). A priori, o conjunto de solução de um método multi-objetivo, será uma aproximação para a FPO (Fronteira de Pareto Ótima) do problema tratado. Portanto, para medir a qualidade de um método multi-objetivo, bastaria averiguar o quão bem o conjunto de soluções obtida aproxima a FPO.

A Figura 21 ilustra o problema sobre a aproximação da FPO. As Figuras 21(a), 21(b) e 21(c) ilustram 3 conjuntos de solução distintos,  $\mathbb{A}$ ,  $\mathbb{B}$  e  $\mathbb{C}$  respectivamente. Vários aspectos podem ser considerados para aferir a qualidade da aproximação.

A Figura 22 ilustra outra questão, mesmo que todas as soluções de um conjunto estejam sobre a FPO, o espalhamento destas é outra medida de qualidade. Este espalhamento pode ser medido por diferentes aspectos. Portanto, vários indicadores de qualidade devem

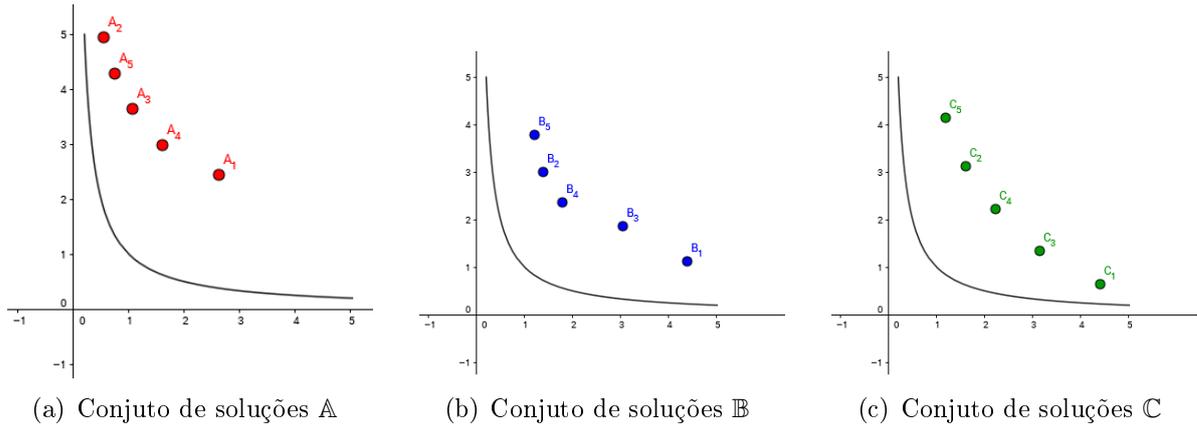


Figura 21: Ilustração da aproximação da FPO pelos conjuntos de solução

ser considerados simultaneamente para aferir qualidade à um conjunto de soluções. Em Zhou et al. (2011) é exibida uma tabela com vários indicadores utilizados na literatura.

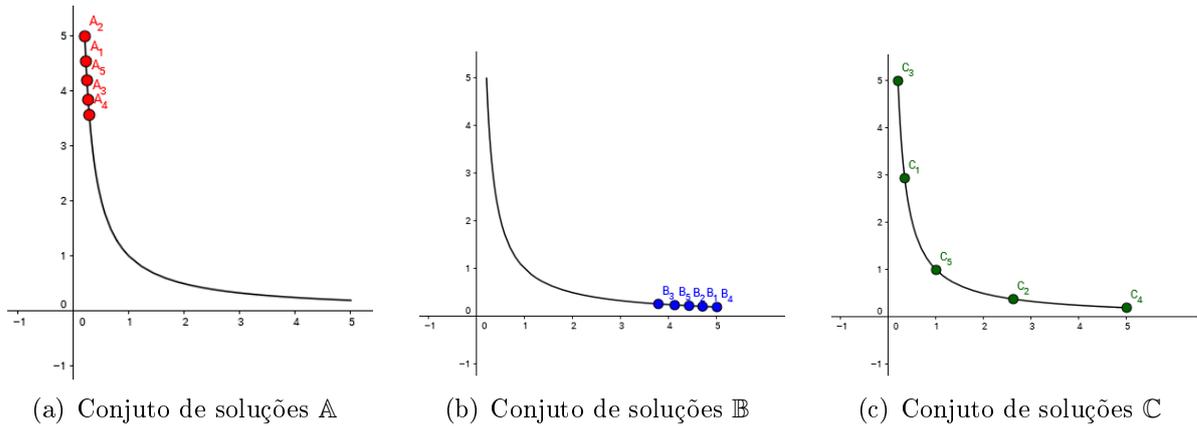


Figura 22: Ilustração de conjuntos de solução possível

Quando a FPO não é conhecida, é estabelecida uma relação entre conjuntos de solução. Em Zitzler et al. (2003) várias relações são enumeradas e analisadas. Para esta Dissertação, foi utilizada a relação de melhor que ( $\triangleright$ ) dada pela Definição 3.

**Definição 3.** *Sejam  $\mathbb{A}$  e  $\mathbb{B}$  dois conjuntos de solução para um problema multi-objetivo. O conjunto  $\mathbb{A}$  é **melhor que** o conjunto  $\mathbb{B}$ ,  $\mathbb{A} \triangleright \mathbb{B}$ , se, e somente se,  $\forall B \in \mathbb{B}, \exists A \in \mathbb{A}$  tal que  $A \prec B$ .*

Pode-se utilizar o indicador de hipervolume, proposto por Zitzler e Thiele (1998) com o nome de tamanho do espaço coberto (*size of the space covered*). O hipervolume  $I_{\overline{H}}(\mathbb{A})$  de um conjunto de soluções  $\mathbb{A}$ , é o volume da região de  $Z$  dominada pelas soluções do conjunto  $\mathbb{A}$ , medida até um ponto de referência  $z_{ref}$ .

**Definição 4.** *Sejam  $z$  e  $z^{Ref} \in Z$  dois pontos no espaço de objetivos  $Z$ , de um problema*

com  $d$  objetivos. Define-se a **região dominada** por  $z$ , com relação à referência  $z^{Ref}$ , como  $D(z, z^{Ref}) \doteq \{z \in Z | z \leq z^{Ref}\}$ , em que  $z \leq z^{Ref} \Leftrightarrow z_i \leq z_i^{Ref}, i = 1, \dots, d$ .

A região dominada por um conjunto de soluções  $\mathbb{A}$ , com relação à uma referência  $z^{Ref}$ , é dada pela expressão 5.1. O hipervolume é o volume da região  $\hat{Z}(\mathbb{A}, z^{Ref})$ , canonicamente estendida do caso bidimensional. A Figura 23 ilustra o hipervolume. Na Figura 23(a) é apresentado um exemplo bidimensional. Um exemplo de hipervolume para um caso tridimensional é exibido na Figura 23(b).

$$\hat{Z}(\mathbb{A}, z^{Ref}) \doteq \cup_{A \in \mathbb{A}} D(A, z^{Ref}) \quad (5.1)$$

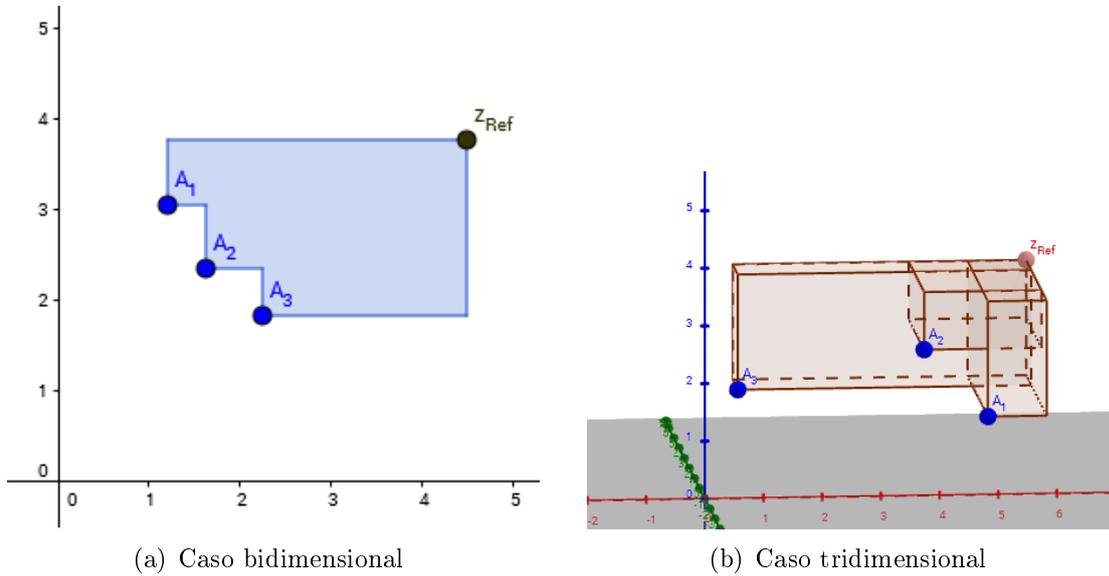


Figura 23: Exemplos de hipervolume

Dentre todos os indicadores unários, somente o hipervolume ( $I_{\bar{H}, z^{Ref}}$ ), proposto por Zitzler e Thiele (1998), preserva a relação dada por  $\triangleright$ . Caso  $I_{\bar{H}, z^{Ref}}(\mathbb{A}) > I_{\bar{H}, z^{Ref}}(\mathbb{B})$ , tem-se que  $\mathbb{A} \triangleright \mathbb{B}$  e  $\mathbb{B} \not\triangleright \mathbb{A}$  (ZITZLER et al., 2003). Portanto, fixando-se um mesmo ponto de referência para cada espaço de objetivos (definido pela instância abordada), o uso do hipervolume mostra-se um indicador confiável.

No Capítulo 3 as incertezas que acompanham a escolha dos pesos para a função objetivo são abordadas brevemente. Como não existe um rigor previamente estabelecido para determinar as concessões, usualmente são propostos alguns conjuntos de pesos para o problema (SUBRAMANIAN et al., 2011) ou pesos baseados nos dados (PHILLIPS et al., 2015).

Trabalhos da literatura que tratam o PAAS, lidam com incertezas sobre a escolha

dos pesos: em Nascimento et al. (2005) são propostos dois conjuntos de pesos, um para cada conjunto de instâncias de teste; em Phillips et al. (2015) os pesos propostos são proporcionais aos dados; em Subramanian et al. (2011) são propostos dois cenários de pesos para os oito objetivos do modelo.

A preocupação com o ajuste dos pesos, reflete a incerteza que os responsáveis por modelar o problema possuem sobre como tratar a incomensurabilidade dos objetivos. Assim uma abordagem multi-objetivo para o PAAS mostra-se promissora, pois pode apresentar uma forma de contornar esta incerteza.

## 5.1 Algoritmo de Busca em Tabelas

O Algoritmo de Busca em Tabelas (ABT) é proposto para tratar o PAAS com abordagem multi-objetivo. O ABT é inspirado no MEAN (*Multiobjective Evolutionary Algorithm with Node-depth encoding*), proposto por Santos et al. (2010a), para o problema de reconfiguração de sistema elétrico. Como o MEAN obteve bons resultados para um problema com 6 objetivos, mostrou-se promissor para tratar o PAAS com 5 objetivos.

O MEAN é baseado em tabelas  $Tab$ , em que cada objetivo  $i = 1, \dots, d$  possui uma tabela  $Tab_i$  com  $SP_i$  soluções. Além destas  $d$  tabelas, é também adicionada uma tabela  $Tab_{d+1}$  para  $F$  função ponderada, ou *aggregation function* (SANTOS et al., 2010a). As tabelas são iniciadas com  $SP_i$  soluções,  $i = 1, \dots, d + 1$ , e ordenadas segundo  $f_i$ , em que  $f_{d+1} = F$ .

Em cada iteração uma tabela  $Tab_i$  é escolhida estocasticamente e uma solução  $X \in Tab_i$  é selecionada aleatoriamente. Um operador é então aplicado sobre  $X$ , levando-a uma nova solução  $X'$ , que será testada em todas as tabelas  $Tab_i$ . Caso o valor de  $f_i(X')$  seja melhor que a pior solução da tabela  $Tab_i$ ,  $X'$  é colocado na tabela  $Tab_i$  e retira-se a pior solução.

Este processo continua até que algum critério de parada seja atingido, originalmente proposto o número máximo de iterações. Cada solução pode estar em mais de uma tabela simultaneamente. Ao fim das iterações, as melhores soluções em cada tabela formam o conjunto de soluções do método.

Para o PAAS modelado, existe alguma diretriz sobre a grandeza dos objetivos, mas não há uma quantificação exata. Assim o MEAN mostra-se novamente promissor, pois pode-se adicionar uma direção de interesse no espaço de objetivos, possibilitando encontrar

soluções não-dominadas próximas à esta direção.

O ABT proposto irá se basear em tabelas, adicionando uma para cada  $f_i$  objetivo  $i = 1, \dots, d$ , bem como uma tabela para cada direção de interesse  $w^j$ ,  $j = 1, \dots, N$ . Adiciona-se uma tabela de não-dominadas  $Tab_{ND}$ , das soluções não-dominadas. O ABT possuirá  $d + N + 1$  tabelas, cada uma será ordenada segundo a função  $f_i(X)$ , para  $i = 1, \dots, d$  ou  $F_j(X)$  (expressão (5.2)) em cada direção de interesse  $j = 1, \dots, N$ .

$$F_j(X) = \sum_{i=1}^d f_i(X) \cdot w_i^j \quad (5.2)$$

As tabelas do ABT serão populadas inicialmente utilizando a heurística construtiva do GRASP, Algoritmo 4.7 na Seção 4.4 do Capítulo 4, com o parâmetro  $\mathbb{L} = 2$  e ordenação aleatória das aulas  $A$ , em cada construção. O número de soluções nas tabelas é limitado pelo parâmetro  $TT$ , portanto em cada tabela constrói-se  $TT$  soluções factíveis distintas.

Quando as tabelas forem iniciadas, também se inicia a tabela de não-dominada  $Tab_{ND}$ , adicionado nesta tabela as soluções não-dominadas dentre todas as tabelas, i.e.,  $Tab_{ND} = \varphi(\cup_{k=1}^{d+N} Tab_k)$ . A tabela  $Tab_{ND}$  não está limitada pelo número máximo de soluções  $TT$ . Para obter a FP deste conjunto de soluções, foi-se utilizado uma modificação do algoritmo de Mishra-Harit proposto em Mishra e Harit (2010).

---

**Algoritmo 5.1:** Pseudo-código do Algoritmo de Mishra-Harit Modificado

---

**Entrada:**  $M$  número de objetivos,  $\mathbb{Y}$  conjunto de soluções,  $ND$  conjunto de soluções não-dominadas

**Saída:**  $\varphi(\mathbb{Y} \cup ND)$

- 1 Ordenar  $\mathbb{Y}$  crescente pelo primeiro objetivo  $f_1$ ;
  - 2 Faça  $\varphi = ND$ ;
  - 3 **para cada**  $Y \in \mathbb{Y}$  **faça**
  - 4     **se**  $\varphi = \emptyset$  **então**
  - 5         Faça  $\varphi = Y$ ;
  - 6     **senão**
  - 7         **para cada**  $X \in \varphi$  **faça**
  - 8             **se**  $Y \prec X$  **então**
  - 9                 Faça  $\varphi = \varphi \setminus \{X\}$ ;
  - 10             **fim**
  - 11         **fim**
  - 12         **se**  $\nexists X \in \varphi | X \prec Y$  **então**
  - 13             Faça  $\varphi = \varphi \cup \{Y\}$ ;
  - 14         **fim**
  - 15     **fim**
  - 16 **fim**
  - 17 **retorna**  $\varphi$
-

O algoritmo de Mishra-Harit parte de um conjunto de soluções  $\mathbb{Y}$ , obtém  $\wp(\mathbb{Y})$  as soluções não-dominadas, com complexidade  $O(d|\mathbb{Y}|^2)$ . O algoritmo é iniciado com uma lista de não-dominadas vazia e iterativamente verifica cada solução de  $\mathbb{Y}$ , adicionando e/ou removendo soluções não-dominadas. A modificação proposta, é feita na iniciação do algoritmo, em que a lista atual de soluções não-dominadas é tomada como lista inicial, na linha 2 do Algoritmo 5.1. Caso  $ND = \emptyset$ , o algoritmo é idêntico ao proposto por Mishra e Harit (2010).

Na linha 4 do Algoritmo 5.1, é verificado se o conjunto de não-dominadas é vazio. Caso positivo, adiciona-se a solução corrente  $Y$  como não-dominada (linha 5). Caso contrário, retira-se as soluções  $X \in \wp$  que são dominadas por  $Y$  (linha 9). Na linha 13, caso a solução  $Y$  não seja dominada por nenhuma solução  $X \in \wp$ ,  $Y$  é adicionada à lista de não-dominadas. O processo termina quando todas as soluções  $Y \in \mathbb{Y}$  foram consideradas.

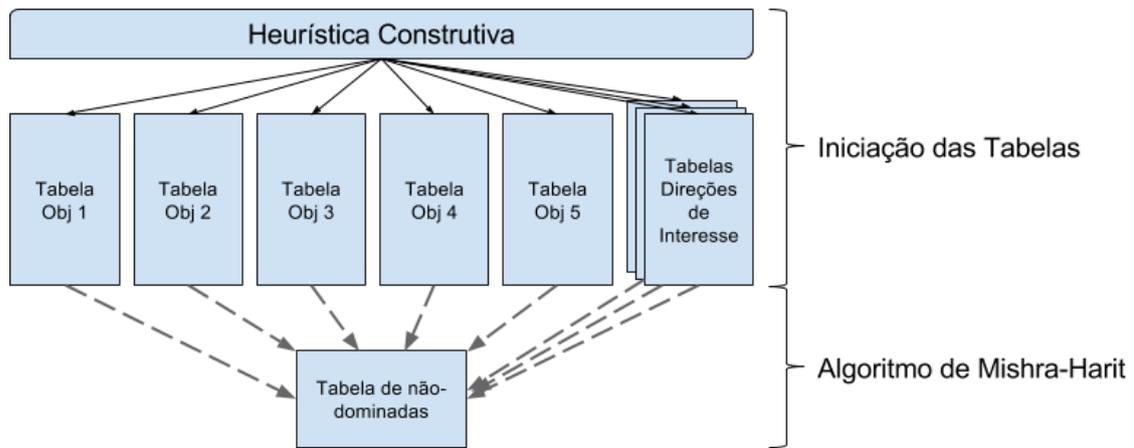


Figura 24: Processo de Iniciação das Tabelas.

A Figura 24 ilustra o processo de iniciação das tabelas, com as etapas principais destacadas à direita. São geradas  $TT$  soluções factíveis para cada tabela de objetivo e de direção de interesse, usando o Algoritmo 4.6 com  $\mathbb{L} = 2$ . A iniciação da tabela  $Tab_{ND}$  utiliza o algoritmo de Mishra-Harit sem modificação, ou seja, na linha 2 do Algoritmo 5.1,  $ND = \emptyset$ .

A busca para o ABT será baseada nas buscas e métodos mono-objetivo do Capítulo 4. Estas buscas partem de uma solução inicial  $X$ , uma estrutura de vizinhança  $\aleph(X)$  e uma direção de busca  $F(X)$ . O algoritmo inicia com a solução corrente  $X' = X$ , em cada iteração obtém-se, aplicando a estrutura de vizinhança  $\aleph(X')$ , um vizinho  $\hat{X}$  de  $X'$ . Caso  $F(\hat{X})$  seja melhor que  $F(X')$ , a solução corrente é atualizada, fazendo  $X' = \hat{X}$ .

Este processo continua até que o critério de parada seja atingido. Ao fim da busca mono-objetivo, a melhor solução obtida durante todas as iterações será a saída do método.

Para as buscas do ABT, a saída da busca será um conjunto de soluções  $\mathbb{Y}$ . Cada vez que a solução corrente da busca for atualizada, é adicionada em  $\mathbb{Y}$ .

Foram propostos para o ABT três estratégias de busca: uma baseada na busca local (Seção 4.2), outra na Busca Tabu (Seção 4.3) e outra no Algoritmo Genético Compacto (Seção 4.6). Cada busca no ABT partirá de uma tabela  $Tab$  e uma direção  $d$ . Para a busca local e BT, uma solução é amostrada aleatoriamente de  $Tab$  e é tomada como solução inicial da busca, enquanto o AGC considera todas as soluções da tabela simultaneamente.

Para a aplicação do AGC, é extraído um modelo probabilístico inicial  $P^0$  da tabela considerada, utilizando a codificação de solução proposta para o AG e o AGC. Esta solução  $M$ , será uma matriz binária com o número de linhas igual ao número de aulas e o número de colunas igual o número de salas. Caso o elemento  $M_{as} = 1$ , sabe-se que a aula  $a$  esta alocada na sala  $s$  nesta solução.

Seja, portanto,  $\mathbb{M}$  a população de soluções na Tabela considerada. O modelo probabilístico será uma matriz real, com valores entre 0 e 1,  $P^0 \in [0, 1]^{|A| \times |S|}$ , com  $A$  conjunto de aulas e  $S$  conjunto de salas. O modelo probabilístico inicial  $P^0$  será iniciado segundo a expressão (5.3).

$$P_{as}^0 = \frac{1}{|\mathbb{M}|} \sum_{M(X) \in \mathbb{M}} M(X)_{as}, \quad a \in A, \quad s \in S \quad (5.3)$$

O ABT é baseado em buscas no espaço de objetivos. Cada tabela  $Tab$  possui uma direção associada à esla  $\vec{D}$ , exceto a  $Tab_{ND}$  que não possui direção associada. Para as tabelas de objetivo, a direção  $\vec{D}$  será o versor canônico do eixo associado ao objetivo. Para as tabelas de direção de interesse, a direção  $\vec{D}$  é dada como entrada no problema, pelos pesos. A função de avaliação de direção  $F(X, \vec{D})$  é definida pela expressão (5.4), em que  $\vec{D}_i$  é o  $i$ -ésimo componente da direção  $\vec{D}$ .

$$F(X, \vec{D}) = \sum_{i=0}^d f_i(X) \vec{D}_i \quad (5.4)$$

Os parâmetros do AGC,  $Pop$  e  $Ordem(A)$ , são fixados em  $Pop = 3700$  e  $Ordem(A)$  por demanda, esta escolha é discorrida na Seção 6.4, do Capítulo 6. Portanto os parâmetros para o AGC como busca do ABT serão a Tabela  $Tab$  de partida, uma direção de busca  $\vec{D}_b$  e  $t_{max}$  tempo máximo de busca. O Algoritmo 5.2 ilustra o processo.

Na linha 5 do Algoritmo 5.2, a população  $\mathbb{M}$  da Tabela  $Tab$  é utilizada para obter

**Algoritmo 5.2:** Pseudocódigo da busca do ABT usando o AGC.

---

**Entrada:**  $Tab$  Tabela,  $\vec{D}_b$  Direção de Busca,  $t_{max}$  Tempo total de execução  
**Saída:** Conjunto de Soluções  $\mathbb{Y}$

- 1 Fazer  $t =$  tempo inicial;
- 2 Fazer  $t_f = t + t_{max}$  tempo final;
- 3 Fazer  $\mathbb{Y} = \emptyset$  conjunto de soluções;
- 4 Fazer  $M$  melhor solução;
- 5 Gerar modelo probabilístico inicial  $P^0$  pela Expressão (5.3);
- 6  $i=0$ ;
- 7 **enquanto**  $t < t_f$  **faça**
- 8     **se**  $\xi(P^i) = 0$  (*Expressão (4.12)*) **então**
- 9         **retorna**  $\mathbb{Y}$ ;
- 10     **fim**
- 11     Amostrar  $M^1$  e  $M^2$  pelo Algoritmo 4.9;
- 12     **se**  $F(M^1, \vec{D}_b) < F(M, \vec{D}_b)$  **então**
- 13          $M = M^1$ ;
- 14          $\mathbb{Y} = \mathbb{Y} \cup \{M^1\}$ ;
- 15     **fim**
- 16     **se**  $F(M^2, \vec{D}_b) < F(M, \vec{D}_b)$  **então**
- 17          $M = M^2$ ;
- 18          $\mathbb{Y} = \mathbb{Y} \cup \{M^2\}$ ;
- 19     **fim**
- 20     Realizar torneio com  $M^1$  e  $M^2$ ;
- 21      $i = i + 1$ ;
- 22     Atualizar o modelo probabilístico  $P^i$  pela Expressão (4.11);
- 23     Fazer  $t =$  tempo atual;
- 24 **fim**
- 25 **retorna**  $\mathbb{Y}$ ;

---

o modelo probabilístico inicial, como indica a expressão (5.3). Nas linhas 12 e 16, as soluções são comparadas segundo a função avaliação de direção, dada pela expressão (5.4). As linhas 14 e 18 são as modificações propostas para obter mais que uma solução em cada busca. O restante do processo é análogo ao AGC mono-objetivo da Seção 4.6 do Capítulo 4.

Em cada iteração do ABT, é escolhida uma tabela e uma solução aleatoriamente, e uma busca é realizada, obtendo  $\mathbb{Y}$  conjunto de soluções. O conjunto  $\mathbb{Y}$ , obtido na busca, é testado contra todas as tabelas, incluindo a tabela de não-dominadas. O teste determina quais elementos de  $\mathbb{Y}$  entrarão na tabela considerada.

Caso  $\mathbb{Y}$ , seja testada contra uma tabela de objetivo ou de direção de interesse, é tomada a direção  $\vec{D}$  desta tabela e  $X^- \doteq \arg \min_{X \in Tab} \{F(X, \vec{D})\}$  a pior solução da mesma. Se a solução  $Y \in \mathbb{Y}$ , é tal que  $F(Y, \vec{D}) < F(X^-, \vec{D})$ , então  $Y$  substitui  $X^-$  na população da tabela  $Tab$ . Quando todas as soluções  $Y \in \mathbb{Y}$  forem consideradas, o teste

acaba. Para a tabela de não dominadas é utilizado o Algoritmo 5.1, com  $ND = Tab_{ND}$ .

O critério de parada adotado foi o tempo máximo de execução do algoritmo. A saída do ABT será a tabela  $Tab_{ND}$ , a tabela das soluções não-dominadas encontradas pelo algoritmo, ou seja, uma aproximação para a FPO do problema. O Algoritmo 5.3 ilustra o ABT.

---

**Algoritmo 5.3:** Pseudocódigo do ABT.

---

**Entrada:**  $d$  Numero de Objetivos,  $TT$  Tamanho das Tabelas,  $N$  Direções de Interesse:  
 $w = \{w_1, \dots, w_N\}$ ,  $B$  Busca,  $t_{max}$  Tempo total de execução,  $t_b$  Tempo de busca

**Saída:** Conjunto de Soluções Não-Dominadas  $\mathbb{Y}$

```

1 Fazer  $t =$  tempo inicial;
2 Fazer  $t_f = t + t_{max}$  tempo final;
3 Criar as  $d + N$  tabelas  $Tab_k$ ,  $k = 1, \dots, d + N$ ;
4 Criar a tabela  $Tab_{ND} = \emptyset$ ;
5 Iniciar Tabelas (Figura 24) ;
6 enquanto  $t < t_f$  faça
7   Selecionar uma Tabela  $Tab$  aleatoriamente;
8   Selecionar uma Direção  $\vec{D}$  aleatoriamente;
9    $\mathbb{Y} = B(Tab, \vec{D}, t_b)$  (Algoritmos 5.2 ou A.1 ou A.2);
10  Testar  $\mathbb{Y}$  contra  $Tab_k$ ,  $k = 1, \dots, d + N$ ;
11  Atualizar  $Tab_{ND}$  com o Algoritmo 5.1;
12  Fazer  $t =$  tempo atual;
13 fim
14 retorna  $Tab_{ND}$ ;
```

---

Os parâmetros de entrada do Algoritmo são as  $N$  direções de interesse, dadas por  $w = \{w_1, \dots, w_N\}$ , o número de soluções nas tabelas  $TT$ , o número de objetivos do problema  $d$  e o tempo total de execução  $t_{max}$ . Também é preciso indicar a busca a ser utilizada  $B$  com o tempo de busca  $t_b$ . Estas Buscas podem ser o AGC, como proposto pelo Algoritmo 5.2, busca local ou Busca Tabu modificadas, Algoritmos A.1 e A.2 respectivamente, descritas no Apêndice A.

As buscas retornam um conjunto de soluções  $\mathbb{Y}$  e possuem a mesma chamada, nominalmente  $B(Tab, \vec{D}, t_b)$ , ou seja, dependem somente da escolha da  $Tab$ , direção de busca  $\vec{D}$  e o tempo de busca  $t_b$ . Portanto a linha 9 do Algoritmo 5.3 esta bem definida para qualquer uma das três buscas utilizadas.

Na linha 5 do Algoritmo 5.3, as tabelas são iniciadas como é indicado pela Figura 24, i.e., cada tabela de objetivo ou direção de interesse possuirão  $TT$  soluções distintas, enquanto  $Tab_{ND}$  é iniciada pelo Algoritmo 5.1 com  $ND = \emptyset$ . Na linha 7, a tabela  $Tab_{ND}$  também pode ser escolhida, porém na linha 8 somente as direções dos objetivos ou de

interesse podem ser selecionadas.

O conjunto de soluções  $\mathbb{Y}$  na linha 10 obtido pela busca escolhida na linha 9, é testado contra todas as tabelas. Para a atualização da tabela  $Tab_{ND}$  na linha 11, é utilizado o Algoritmo 5.1, com  $ND = Tab_{ND}$ . O critério de parada será o tempo total de execução  $t_{max}$  e a saída do método é a tabela de não-dominados obtida,  $Tab_{ND}$ .



---

---

# CAPÍTULO 6

---

## TESTES COMPUTACIONAIS

### 6.1 Ambiente de Testes

As implementações dos modelos propostos foi feita no Visual Studio 2008 utilizando a linguagem C++, com o concert da ferramenta comercial IBM ILOG CPLEX 12.6. Para os algoritmos implementados foi utilizada a linguagem Java, com a IDE NetBeans 8.0.1. O modelo com cinco métricas, Seção 3.2, é um caso generalizado do modelo com duas métricas, Seção 3.1. Portanto, somente o modelo com cinco métricas foi abordado nesta dissertação.

Os testes de ajuste de parâmetros dos algoritmos mono-objetivo, da Seção 6.4, foram executados em uma rede Fedora Linux de 8 computadores i7 com 16 GB de RAM. Os testes de desempenho mono-objetivo (Seção 6.5), multi-objetivo (Seção 6.6) e o método exato (Seção 6.3), foram executados com a utilização dos recursos computacionais do Centro de Ciências Matemáticas Aplicadas à Indústria (CeMEAI) financiados pela FAPESP.

### 6.2 Geração dos Dados

O trabalho trata de um estudo de caso no ICMC, onde foi desenvolvida a Dissertação. Porém, ainda assim houve dificuldades para obtenção dos dados. Portanto somente duas instâncias reais foram obtidas, cada uma correspondendo a um semestre letivo. Estas

instâncias, **ICMC-2014-2** e **ICMC-2015-1**, são relativas ao segundo semestre letivo de 2014 e o primeiro semestre letivo de 2015, respectivamente.

Devido à dificuldade na obtenção dos dados e para a verificação do desempenho das abordagens, foi proposto um gerador de instâncias, baseado nas instâncias reais. O gerador gera instâncias  $n$  vezes maiores que as reais, ou seja, os tamanho dos conjuntos de turma, sala e perfil são multiplicados por  $n$ . O restante dos dados de entrada são ajustados de acordo. Por este gerador, foram amostradas 15 instâncias, 5 para cada fator  $n = 1, 2, 5$ .

Tabela 2: Dimensões utilizadas nas Instâncias

Nome da Instância	H	T	S	R	P
<b>ICMC2014-2</b>	75	141	23	5	30
<b>ICMC2015-1</b>	85	143	23	5	39
<b>1x</b>	85	143	23	5	39
<b>2x</b>	85	246	46	5	78
<b>5x</b>	85	715	115	5	195

A Tabela 2, exibe as dimensões dos conjuntos de horário ( $H$ ), turmas ( $T$ ), salas ( $S$ ), recursos ( $R$ ) e currículos ( $C$ ) para as instâncias reais e as geradas. A Tabela 3 descreve as 17 instâncias utilizadas. A primeira coluna é o nome da instância, as próximas 8 colunas são as densidades (médias) dos dados de entrada (descritos no Capítulo 3).

Tabela 3: Instâncias utilizadas e suas características

Instância	Densidade dos Dados								
	$\rho$	$\sigma$	$\tau$	$\Upsilon$	CAP	TAM	DIS	PVAZ	PRET
<b>ICMC2014-2</b>	0,0467	0,3913	0,3049	0,0373	54,7391	26,6312	19,5917	0,2609	4,8405
<b>ICMC2015-1</b>	0,0400	0,3913	0,2000	0,0256	56,0435	35,4126	19,5917	0,2609	4,7871
<b>1x-1</b>	0,0404	0,4087	0,2000	0,0278	65,2174	36,5804	25,2439	0,1739	5,0457
<b>1x-2</b>	0,0406	0,4174	0,2000	0,0271	57,4783	32,1049	25,5652	0,2174	5,0981
<b>1x-3</b>	0,0407	0,4174	0,2000	0,0271	63,9565	34,5664	25,1342	0,1739	5,0970
<b>1x-4</b>	0,0406	0,4087	0,2000	0,0276	61,8696	35,2797	23,7732	0,3478	4,9978
<b>1x-5</b>	0,0411	0,4000	0,2000	0,0271	59,1304	34,1399	24,1550	0,2609	5,1215
<b>2x-1</b>	0,0411	0,3957	0,2000	0,0135	54,2174	32,8217	25,3185	0,3261	5,0691
<b>2x-2</b>	0,0392	0,3783	0,2000	0,0137	55,8913	35,2483	24,4301	0,2826	5,0471
<b>2x-3</b>	0,0397	0,3913	0,2000	0,0139	55,4783	36,3427	24,1503	0,2391	5,1037
<b>2x-4</b>	0,0406	0,4304	0,2000	0,0141	63,0652	35,4860	25,1853	0,1304	5,0348
<b>2x-5</b>	0,0386	0,3739	0,2000	0,0142	54,7609	35,5874	24,5435	0,3261	5,0624
<b>5x-1</b>	0,0396	0,3983	0,2000	0,0054	54,6870	34,5427	25,0507	0,2609	5,0574
<b>5x-2</b>	0,0397	0,3757	0,2000	0,0054	55,7217	36,2643	25,4811	0,2783	5,0452
<b>5x-3</b>	0,0402	0,3878	0,2000	0,0055	56,6870	36,5944	25,4864	0,2957	5,0580
<b>5x-4</b>	0,0398	0,3896	0,2000	0,0054	54,3130	35,7958	25,2508	0,2522	5,0448
<b>5x-5</b>	0,0404	0,3826	0,2000	0,0056	55,2696	35,9189	25,5974	0,2696	5,0592

Como visto no Capítulo 3, a função objetivo do modelo depende da escolha dos pesos  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  e  $\epsilon$ . No Capítulo 5, foram apresentados os impactos da escolhas dos pesos para os métodos mono-objetivos. Conjuntamente com os decisores, foram elaborados 3 conjuntos de pesos, conforme indicado na Tabela 4.

Tabela 4: Pesos para as 5 Métricas

Peso	Parâmetros				
	$\alpha$	$\beta$	$\gamma$	$\delta$	$\epsilon$
<b>1</b>	0,1	10000	10	1000	100
<b>2</b>	1	5000	5	2000	500
<b>3</b>	10	20000	100	3000	1000

### 6.3 Resultados do Método Exato

A resolução exata do modelo, por meio da ferramenta comercial ILOG IBM CPLEX 12.6, foi utilizada a fim de comparação com os algoritmos implementados. O CPLEX foi executado por 1 hora para cada instância proposta, a fim de estabelecer um limitante inferior. Para comparações com as abordagens mono-objetivo, foi executado o CPLEX com tempo máximo de 5 minutos.

O CPLEX foi executado 51 vezes com 1 hora e 51 vezes com 5 minutos, uma para cada combinação de instância (Tabela 3) e peso (Tabela 4). Os resultados para cada instância e peso são apresentados na Tabela 5. A coluna  $\hat{F}(i, p)$ , é o limitante inferior da instância  $i$  com o peso  $p$ , obtido com 1 hora de execução. A coluna  $F_{5m}^C(i, p)$  é a melhor solução obtida com 5 minutos de execução do CPLEX, enquanto  $F_{1h}^C(i, p)$  é a melhor solução obtida com 1 hora. As colunas  $GAP_{5m}(i, p) \doteq \frac{F_{5m}^C(i, p) - \hat{F}(i, p)}{F_{5m}^C(i, p)}$  e  $GAP_{1h}(i, p) \doteq \frac{F_{1h}^C(i, p) - \hat{F}(i, p)}{F_{1h}^C(i, p)}$  são o GAP de otimalidade para 5 minutos e 1 hora respectivamente. O símbolo  $\bullet$  indica quando não houve solução.

Foi feita a opção por executar o CPLEX com o tempo máximo de 5 minutos, a fim de possibilitar comparações com os algoritmos propostos. Destaca-se porém que o CPLEX e os métodos foram implementados em linguagens de programação distintas, C++ e JAVA, respectivamente. A diferença de performance entre o JAVA e o C++ vem caindo ao longo dos anos, dado principalmente pelo avanço no gerenciamento do Coletor de Lixo (Garbage Collector) da JVM (JAVA Virtual Machine) (HUNDT, 2011), porém o desempenho do C++ é superior.

O CPLEX não se mostrou muito eficiente para lidar com o tempo fixado em 5 minutos, com as instâncias maiores, já que em nenhuma das instâncias  $5x$  foi encontrada solução factível. Percebe-se também que o GAP de otimalidade cresce rapidamente com o tamanho da instância, como ilustra a Figura 25. Na Figura 25, nota-se que as instâncias reais e  $1x$  possuem GAP de otimalidade baixo, com o máximo em 0,2309. Porém, para as instâncias  $2x$  o GAP mínimo foi 0,7067.

Com uma hora de execução, o CPLEX consegue obter soluções factíveis para todas

Tabela 5: Soluções obtidas com o CPLEX

Instância	Peso	$\hat{F}(i, p)$	$F_{1h}^C(i, p)$	$GAP_{1h}$	$F_{5m}^C(i, p)$	$GAP_{5m}$
<b>ICMC-2014-2</b>	1	23517,1	23517,1	0,0000	23517,1	0,0000
	2	111511	111511	0,0000	111511	0,0000
	3	346500	346500	0,0000	346500	0,0000
<b>ICMC-2015-1</b>	1	17600,8	17600,8	0,0000	17600,8	0,0000
	2	92187	92187	0,0000	92187	0,0000
	3	283970	283970	0,0000	283970	0,0000
<b>1x-1</b>	1	33091	36649,7	0,0971	40838,3	0,1897
	2	155850	159503	0,0229	171002	0,0886
	3	412907	491030	0,1591	536890	0,2309
<b>1x-2</b>	1	43532,9	47603	0,0855	51127,1	0,1485
	2	186821,4	189014	0,0116	203047	0,0799
	3	492989,9	548620	0,1014	581730	0,1525
<b>1x-3</b>	1	32333,6	33067,7	0,0222	34599	0,0655
	2	157963	157963	0,0000	158468	0,0031
	3	424541,8	450490	0,0576	465230	0,0874
<b>1x-4</b>	1	36438,15	37407	0,0259	39563,5	0,0790
	2	157817	157817	0,0000	159675	0,0116
	3	431545,1	457630	0,0570	481560	0,1039
<b>1x-5</b>	1	32312,4	33567,8	0,0374	41603,6	0,2233
	2	160429	160429	0,0000	167825	0,0441
	3	425416,9	473580	0,1017	531970	0,2003
<b>2x-1</b>	1	59175,5679	102753,2	0,4241	744771	0,9205
	2	286772,7024	319916	0,1036	$2,04 \times 10^7$	0,9859
	3	789490,65	1145850	0,3110	2805230	0,7186
<b>2x-2</b>	1	53563,1718	129881,6	0,5876	$2,35 \times 10^7$	0,9977
	2	267890,3304	330484	0,1894	$1,99 \times 10^7$	0,9866
	3	747053,435	1128650	0,3381	$2,23 \times 10^8$	0,9966
<b>2x-3</b>	1	61230,6682	103342,9	0,4075	$1,13 \times 10^7$	0,9946
	2	303979,4688	353136	0,1392	$1,10 \times 10^7$	0,9723
	3	810023,066	1094330	0,2598	2762130	0,7067
<b>2x-4</b>	1	61870,194	452267,5	0,8632	848766,6	0,9271
	2	293056,5712	369368	0,2066	$1,21 \times 10^7$	0,9758
	3	812918,601	3184170	0,7447	$1,18 \times 10^8$	0,9931
<b>2x-5</b>	1	70100,6011	196690,8	0,6436	$1,03 \times 10^7$	0,9932
	2	314518,0113	372961	0,1567	$1,00 \times 10^7$	0,9686
	3	825518,304	1469940	0,4384	$9,22 \times 10^7$	0,9910
<b>5x-1</b>	1	92315,3	$1,45 \times 10^8$	0,9994	•	•
	2	488644	$1,04 \times 10^8$	0,9953	•	•
	3	1487250	$1,42 \times 10^9$	0,9989	•	•
<b>5x-2</b>	1	86027	$1,28 \times 10^8$	0,9993	•	•
	2	453046	$9,31 \times 10^7$	0,9951	•	•
	3	1402210	$1,25 \times 10^9$	0,9989	•	•
<b>5x-3</b>	1	84294,9	$1,34 \times 10^8$	0,9994	•	•
	2	447209	$9,68 \times 10^7$	0,9954	•	•
	3	1388170	$1,31 \times 10^9$	0,9989	•	•
<b>5x-4</b>	1	83701,9	$1,31 \times 10^8$	0,9994	•	•
	2	444017	$9,51 \times 10^7$	0,9953	•	•
	3	1394450	$1,27 \times 10^9$	0,9989	•	•
<b>5x-5</b>	1	91947	$1,41 \times 10^8$	0,9993	•	•
	2	485062	$1,01 \times 10^8$	0,9952	•	•
	3	1472030	$1,37 \times 10^9$	0,9989	•	•

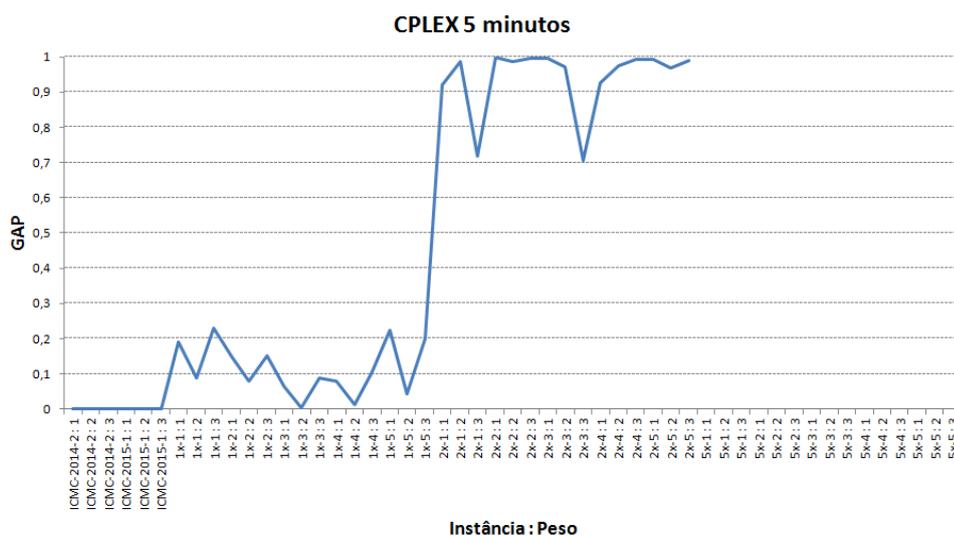


Figura 25: GAP de otimalidade para o CPLEX com 5 minutos comparado com o limitante inferior

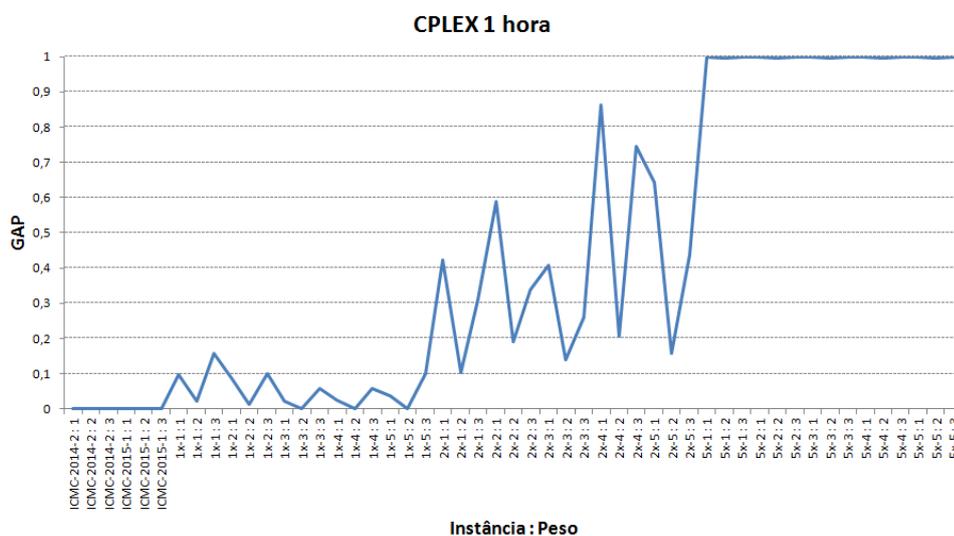


Figura 26: GAP de otimalidade para o CPLEX com 1 hora comparado com o limitante inferior

as combinações de instância e peso, como ilustra a Figura 26. Em algumas combinações obtém-se solução ótima, que não era obtida com 5 minutos, nas instâncias **1x-4** e **1x-5** para o peso 2. Nota-se que o GAP de otimalidade para as instâncias **5x** são muito grande, o menor valor obtido de GAP foi 0,9951.

## 6.4 Testes de Sensibilidade de Parâmetros

Os métodos mono-objetivo propostos no Capítulo 4, dependem da escolha de parâmetros, exceto a busca local que é não-paramétrica. A Tabela 6 sintetiza os parâmetros de cada método investigado. Existem trabalhos na literatura que lidam justamente com a questão dos parâmetros nos métodos, gerando métodos não-paramétricos (BEHESHTI; SHAMSUDDIN, 2015) ou fazendo *tunning* dos parâmetros (EIBEN; SMIT, 2011).

Tabela 6: Parâmetro dos Algoritmos Propostos

Metaheurística	Parâmetros
Algoritmo Genético	$T_r, T_m, Pop$
Algoritmo Genético Compacto	$Pop, Amostragem$
GRASP	$L, N$
GRASP Tabu	$L, N, Tabu$
Busca Tabu	$Tabu$
Busca Local	

Para esta Dissertação, o ajuste dos parâmetros foi realizada utilizando um experimento estatístico. É Investigado o impacto da escolha dos parâmetros na eficiência dos métodos, identificando as melhores configurações de parâmetros para cada método. Para a realização do experimento, os parâmetros são discretizados, conforme ilustra a Tabela 7.

Tabela 7: Discretização dos Parâmetro dos Algoritmos Propostos

Metaheurística	Parâmetros	Nº Comb
Algoritmo Genético	$T_r \in \{0.1, \dots, 1\}, T_m \in \{0.0, \dots, 1\}, Pop \in \{50, \dots, 500\}$	1100
AG Compacto	$Pop \in \{100, \dots, 5000\}, Ordem(A) \in \{0, 1\}$	100
GRASP	$L \in \{2, 3, 4, 5\}, N \in \{10, \dots, 30\}$	20
GRASP Tabu	$L \in \{2, \dots, 5\}, N \in \{10, \dots, 30\}, Tabu \in \{10, \dots, 50\}$	100
Busca Tabu	$Tabu \in \{10, \dots, 50\}$	5
Busca Local		1

A instância **ICMC-2015-1** foi escolhida para a condução dos testes. Para cada combinação de parâmetro e cada peso (Tabela4) foi executado um ensaio com 10 execuções de cada método, com o tempo máximo de execução fixado em 5 minutos.

Cada par instância-peso  $(i, p)$  possui uma ordem de grandeza distinta de função objetivo. É utilizado portanto, o limitante inferior  $\hat{F}(i, p)$  (Tabela 5) para normalizar o valor de  $F_m^j(i, p)$  obtido. Em que  $F_m^j(i, p)$  é o valor de função objetivo (expressão (3.8)) obtida com o método  $m$ , a configuração de parâmetros  $j$ , para a instância  $i$  com peso  $p$ . O GAP de otimalidade é definido pela expressão 6.1, permitindo comparar soluções de instâncias

diferentes.

$$GAP_{i,p,m,j} = \frac{F_m^j(i,p) - \hat{F}(i,p)}{F_m^j(i,p)} \quad (6.1)$$

### 6.4.1 Apresentação dos Resultados

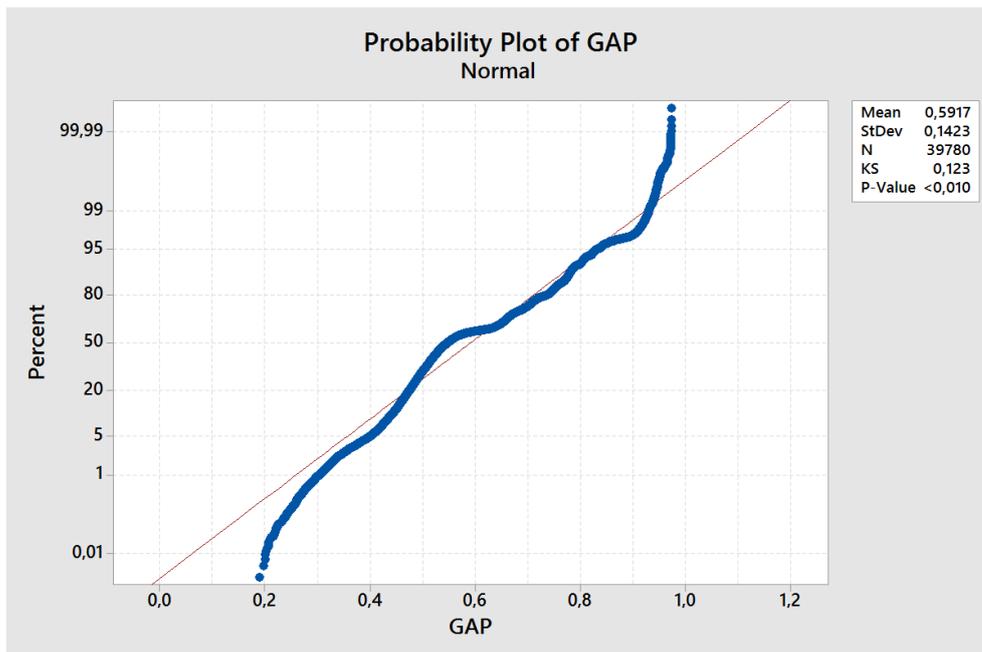
Inicialmente são buscadas evidências nos dados que o GAP, calculado pela expressão 6.1, possua uma distribuição normal para o experimento. A condição de normalidade é obrigatória para utilização de alguns estimadores e desejável para outros (WALPOLE et al., 2012). Foi realizado um teste de normalidade de Shapiro-Walker, como ilustrado pela Figura 27. Porém, não há evidências de normalidade nos dados ( $p\text{-value} < 0,01$ ). As Figuras 27(a) e 27(b) são os gráficos de probabilidade e histograma dos dados, respectivamente.

Como não há evidencia de normalidade nos dados, a utilização de alguns estimadores é restringida, mas é possível aplicar uma Análise de Variância (ANOVA), pois mesmo na ausência da condição de normalidade dos dados, apresentando bons resultados. A ANOVA é baseada no teste de Welch (WELCH, 1947) e a estatística  $t$ , da distribuição T de Student, utilizando os graus de liberdade dados pela equação de Welch-Satterthwaite. A hipótese nula da ANOVA é que todas as médias amostrais são iguais para qualquer fator.

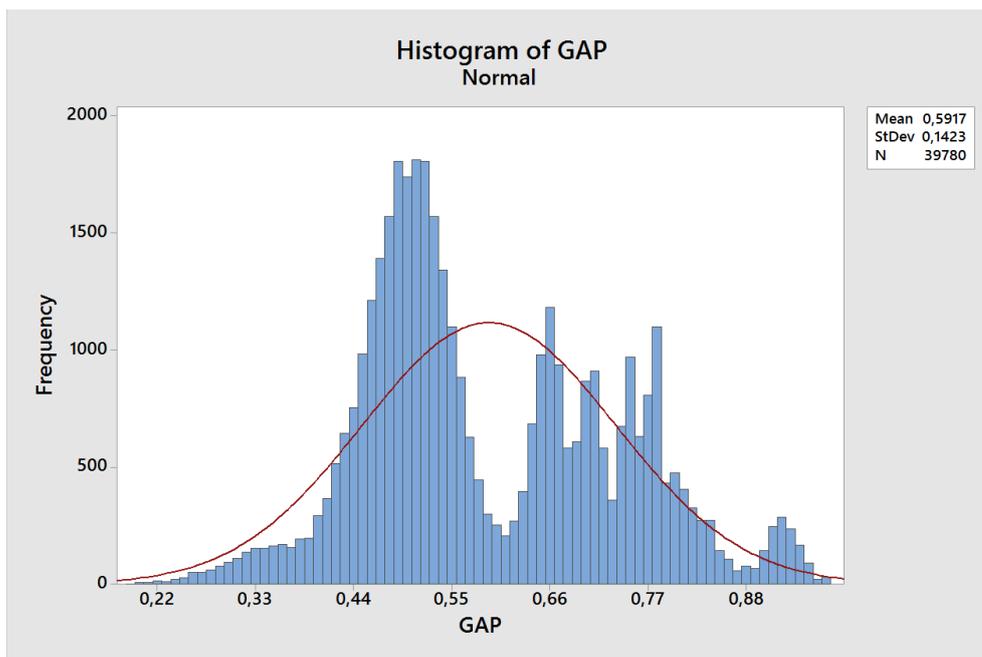
Fazendo uma ANOVA com os pesos como variável categórica e o GAP como variável resposta, como ilustra a Figura 28, é obtido um modelo com baixo valor de  $R^2$  ajustado (56,53%). Portanto somente 56,53% da variância dos dados é ajustada pelo modelo. Entretanto, obtém-se um  $p\text{-value}$  baixo ( $< 0,00001$ ) para o fator peso, indicando que independente do método utilizado, dentre os propostos nesta dissertação, a escolha dos pesos por parte do decisor terá um impacto significativo na eficiência do mesmo.

São analisados em seguida cada método individualmente, de forma a tentar identificar a configuração de parâmetros mais robustas e eficientes. Para cada método, procura-se encontrar um modelo de regressão linear múltipla, para compreender a intensidade com que cada parâmetro afeta o algoritmo. Caso o modelo obtido possua boa aderência aos dados, aplica-se uma análise da superfície de Desejabilidade Composta (*Composite Desirability*) (FITRIANTO; MIDI, 2011), a fim de obter a melhor configuração de parâmetros para o método.

À guisa de exemplo, é utilizado o método GRASP para exemplificar os testes, conduzir as análises e apresentar os resultados. Análises similares foram feitas para todos os



(a) Teste de Normalidade de Shapiro-Wilk



(b) Histograma de GAP

Figura 27: Evidência de Não Normalidade de GAP nos dados

métodos, porém somente os detalhes relevantes de cada um será ressaltado.

O GRASP, descrito na Seção 4.4, só possui 2 parâmetros:  $L$  tamanho da lista elite e  $N$  número de disparos da busca local, portanto exemplificará bem a metodologia utilizada. A fim de caracterizar a amostra, foram feitos os histogramas, agrupando os experimentos por cada nível dos fatores, com uma curva normal ajustada, como pode ser visto na Figura 29.

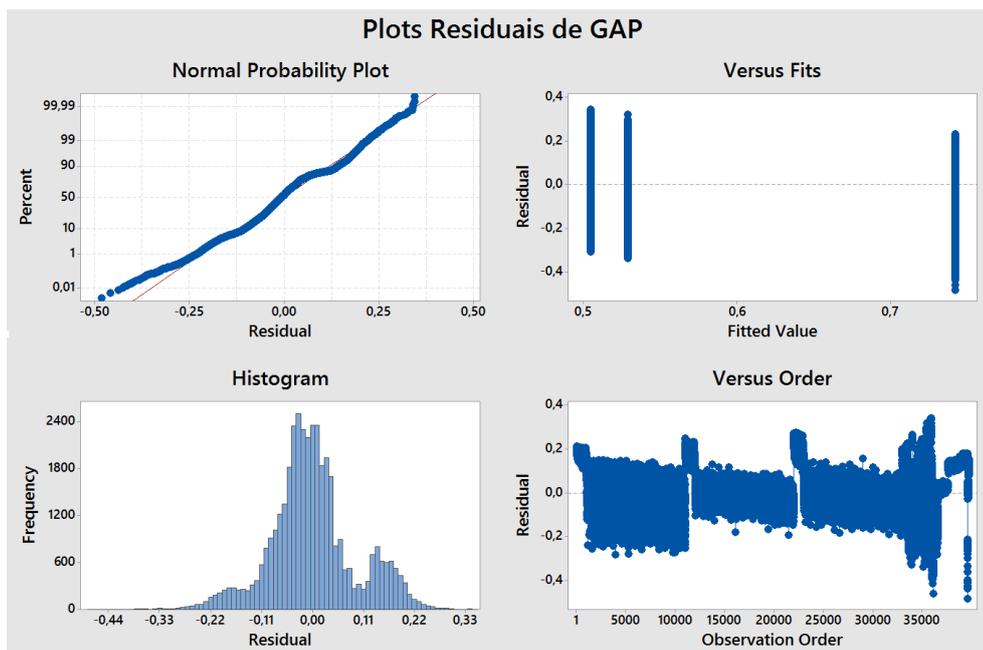


Figura 28: Análise dos Resíduos para o Modelo Ajustado

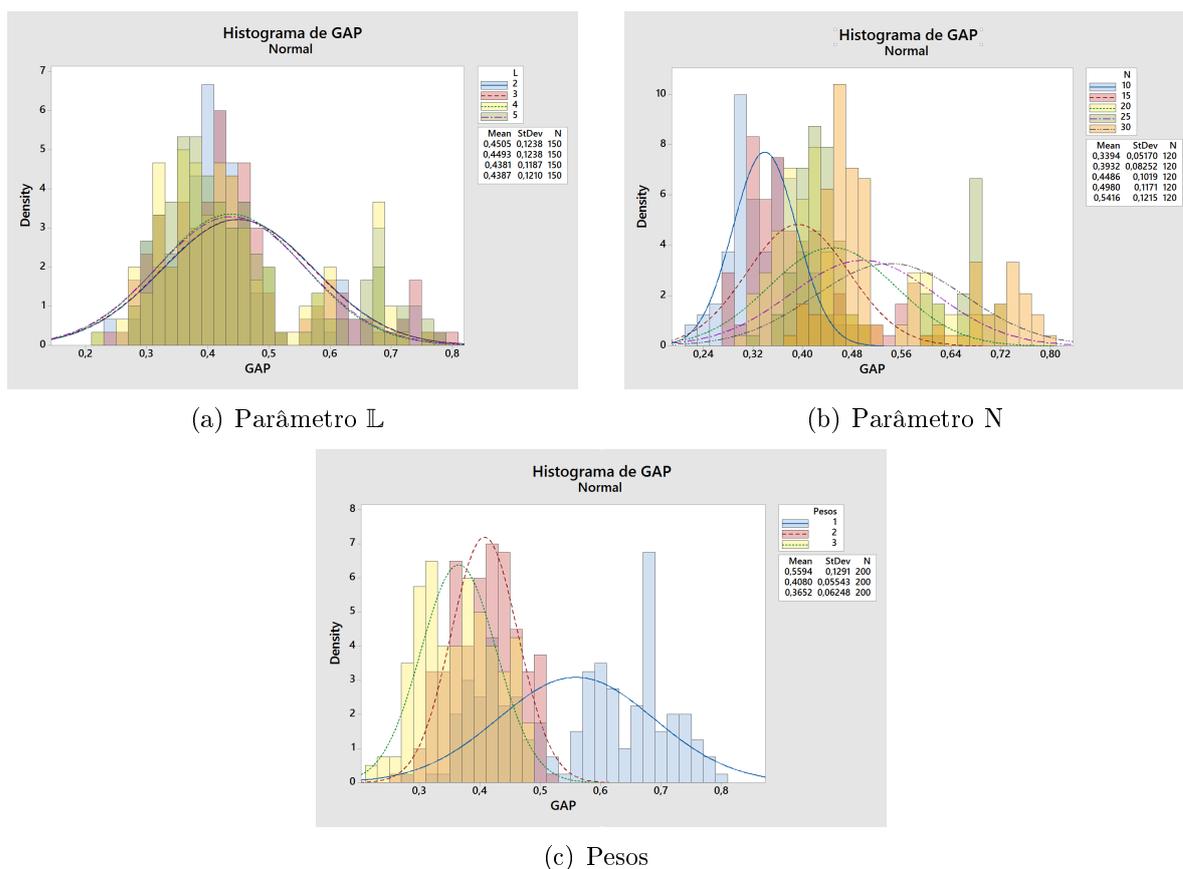


Figura 29: Histogramas de GAP com Normais Ajustadas para o GRASP

É destacado o baixo impacto da escolha do parâmetro  $L$  nos resultados do GRASP, como ilustrado na Figura 29(a). Para o parâmetro  $N$ , no entanto, a variação no GAP foi

maior, como evidenciam as curvas normais ajustadas da Figura 29(b). Apesar do peso não ser um parâmetro ajustável, o histograma ilustrado pela Figura 29(c) demonstra o impacto que a escolha dos pesos acarreta ao método.

O gráfico de efeitos principais, ilustrado para o GRASP na Figura 30, traça a variação da média para cada nível de cada fator. Este gráfico deixa evidente o baixo impacto da escolha do parâmetro  $L$ , enquanto o parâmetro  $N$  causa um grande impacto. Como havia sido discutido, a escolha dos pesos impactam todos os métodos propostos.

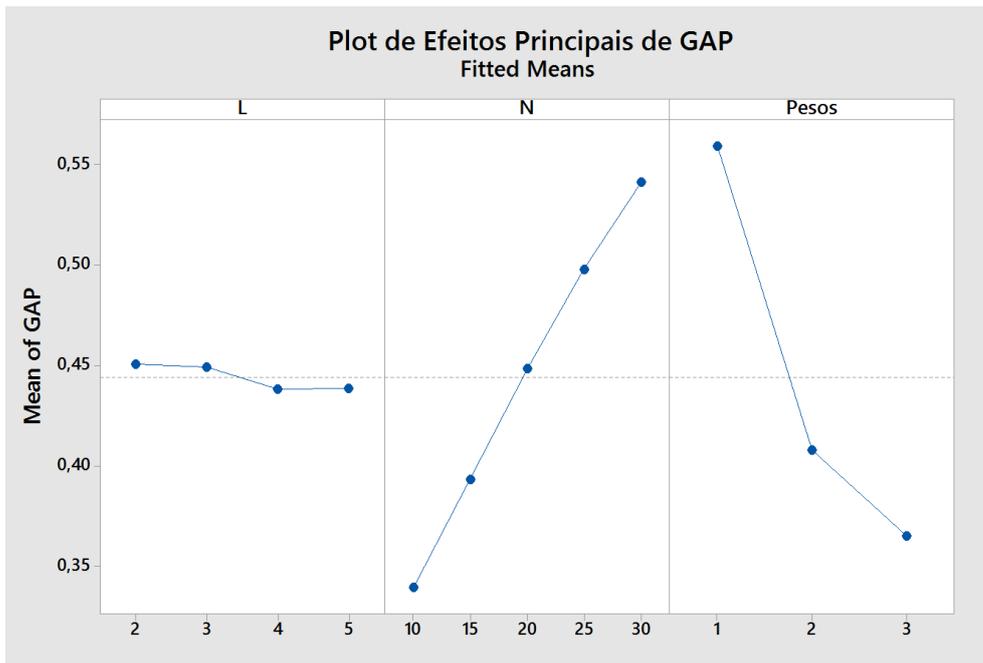


Figura 30: Gráfico de Efeitos Principais de GAP

Considerando os histogramas da Figura 29, é esperado que a variável resposta GAP não apresente distribuição normal, porém um teste de normalidade de Shapiro-Wilk é aplicado para aferir esta condição. A Figura 31 é o gráfico de probabilidade de GAP contra uma normal (reta), pode-se ver que não existe uma boa aderência dos dados à distribuição normal (p-value < 0,010).

No intuito de compreender como os fatores influenciam a variável resposta foi realizada uma regressão linear múltipla em que cada um dos parâmetros é tratado como uma variável categórica. O modelo obtido é dado pela expressão (6.2).

$$\begin{aligned}
 GAP = & 0,44418 + 0,00635L_2 + 0,00512L_3 - 0,00603L_4 - 0,00544L_5 \\
 & - 0,10475N_{10} - 0,05094N_{15} + 0,00446N_{20} + 0,05384N_{25} + 0,09739N_{30} \quad (6.2) \\
 & + 0,11519Pes_1 - 0,03618Pes_2 - 0,07900Pes_3
 \end{aligned}$$

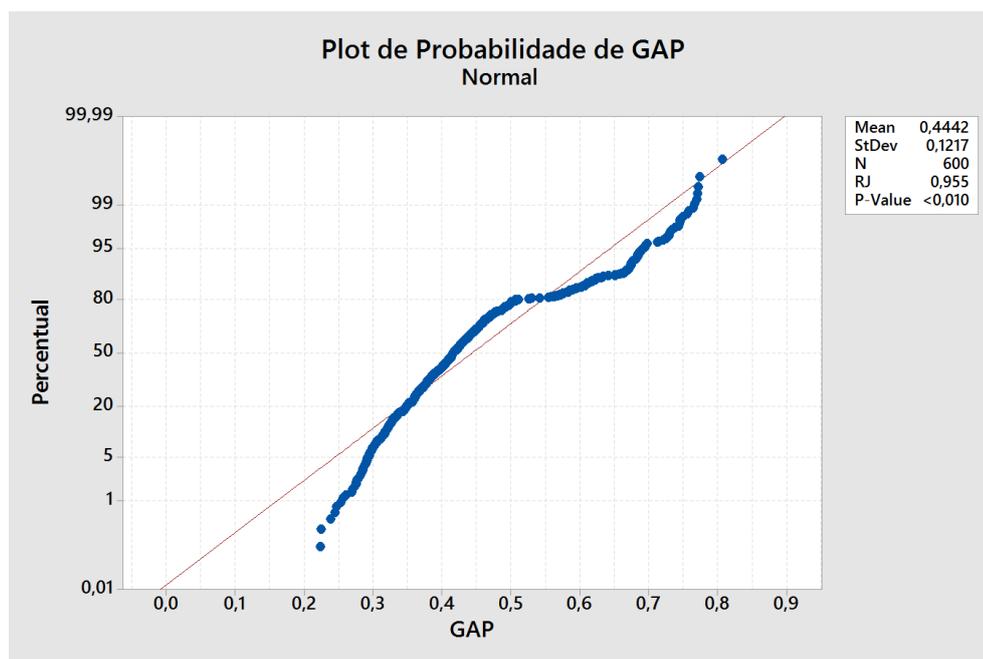


Figura 31: Gráfico de Probabilidade de GAP

Na expressão 6.2 as variáveis  $L_i$  estão identificadas para cada  $L = i$  da discretização proposta, exibida na Tabela 7, bem como  $N_j$  e  $Peso_k$ , e.g.,  $L_2$  é o experimento em que o parâmetro  $L$  foi fixado em 2. Analogamente para os outros parâmetros.

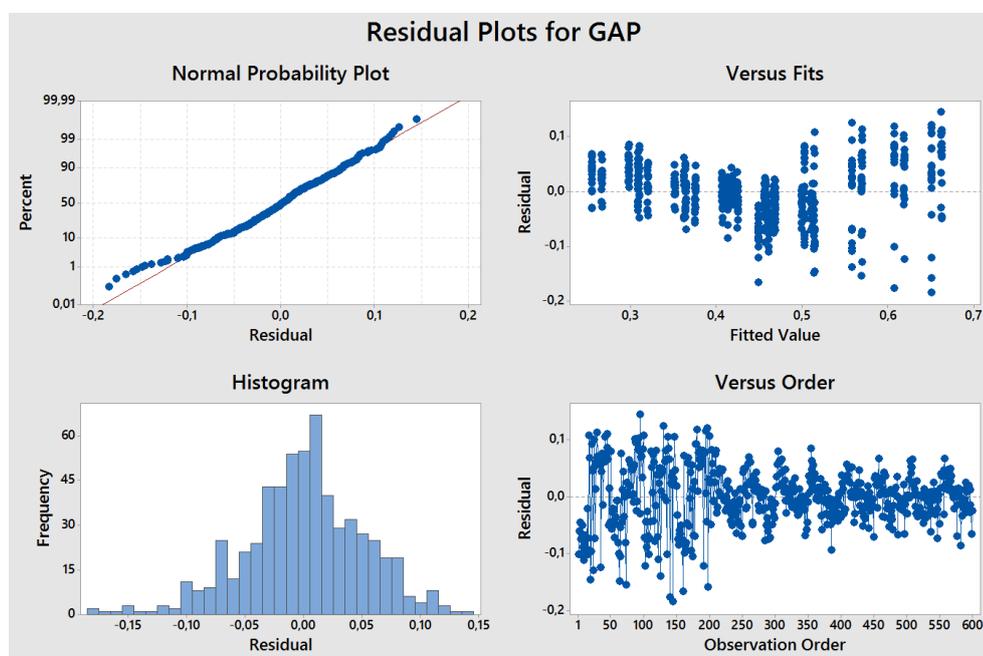


Figura 32: Gráficos da Análise de Resíduos da Regressão Linear Múltipla para o GRASP

A análise de resíduos, ilustrada pela Figura 32, indica que existe uma normalidade nos resíduos. Aliado com o  $R^2 = 82,04\%$  ajustado da regressão linear, há indícios que o modelo é um bom preditor para a variável resposta.

A partir do modelo linear generalizado é possível fazer uma análise de superfície da variável GAP por meio da Função de Desejabilidade Composta, analisando todos os parâmetros simultaneamente. Tendo porém, a escolha de pesos impacto significativo em qualquer método, e estando a mesma a cargo do decisor, deve-se escolher o conjunto de parâmetros mais robustos, ou seja, o par  $(L, N)$  que minimizará o GAP, independente da escolha dos pesos pelo decisor.

Dessa forma, foram realizados os testes a fim de minimizar a variável resposta, i.e., a função de Desejabilidade Composta seja máxima. A Desejabilidade Composta é a média das Desejabilidades individuais, portanto o impacto da escolha dos pesos é reduzido. A Tabela 8 elenca os dez maiores valores da função de Desejabilidade Composta para cada combinação de parâmetros.

Tabela 8: Desejabilidade Composta para as combinações de parâmetros do GRASP

<b>L</b>	<b>N</b>	<b>Peso</b>	<b>GAP Ajustado</b>	<b>Desejabilidade Composta</b>
4	10	3	0,254396	0,947795
5	10	3	0,254986	0,946783
3	10	3	0,265550	0,928651
2	10	3	0,266783	0,926536
4	10	2	0,297215	0,874307
5	10	2	0,297805	0,873295
4	15	3	0,308206	0,855445
3	10	2	0,308370	0,855164
5	15	3	0,308796	0,854432
2	10	2	0,309603	0,853048

Conclui-se portanto a análise de superfície destacando os parâmetros  $L = 4$  e  $N = 10$  como os melhores, deste experimento, para o GRASP. Deve-se destacar também, o baixo impacto da escolha do parâmetro  $L$  para o resultado, visualmente ilustrado pela Figura 29(a).

Os parâmetros do GRASP<sub>T</sub>, apresentado na seção 4.4 do Capítulo 4, são os mesmos do GRASP, acrescido do parâmetro *Tabu*, de tamanho máximo da Lista Tabu da BT. Portanto os parâmetros do GRASP<sub>T</sub> são:  $L$ ,  $N$  e *Tabu*.

Destaca-se, acerca do GRASP<sub>T</sub>, o valor de 96,65% para o  $R^2$  ajustado da regressão linear múltipla, dando indícios de uma boa aderência do modelo aos dados. O baixo impacto da escolha dos parâmetros fica evidenciado pelo gráfico de efeitos principais, Figura 33, sendo este um destaque da análise deste método. O peso escolhido pelo decisor terá papel mais contundente para o método do que a escolha dos parâmetros do mesmo, como

evidencia a Figura 33(a). Porém a escolha dos parâmetros ainda causa efeitos estatisticamente significantes na performance do método, como ilustrado pela Figura 33(b).

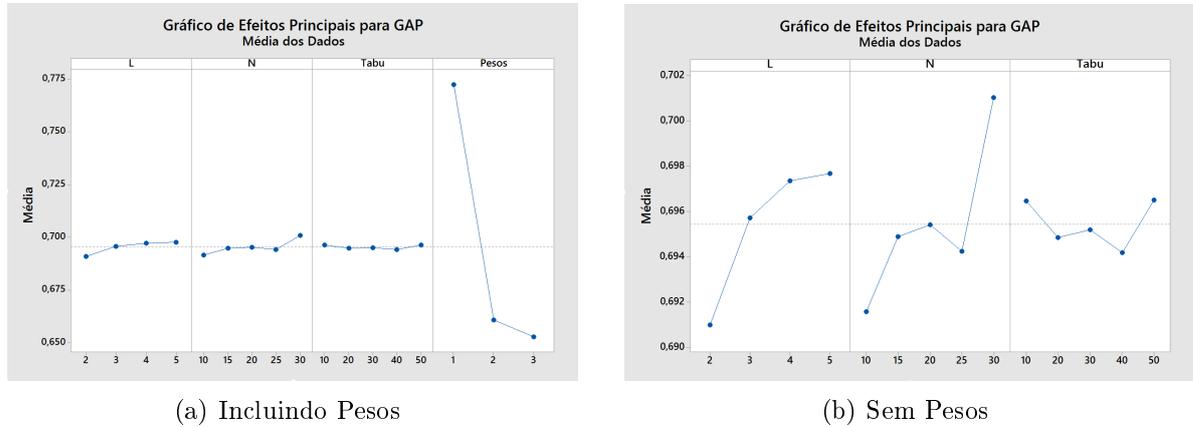


Figura 33: Gráfico de Efeitos Principais sobre as Médias de GAP para o GRASP<sub>T</sub>

Os parâmetros escolhidos, utilizando a função de Desejabilidade Composta, para o GRASP<sub>T</sub> serão portanto:  $L = 2$ ,  $N = 10$ ,  $Tabu = 40$ .

Passando para o Algoritmo Genético Compacto, pela Tabela 6 verifica-se que o AGC possui 2 parâmetros somente:  $Pop$  e  $Ordem(A)$ . O primeiro é o tamanho da população virtual e o segundo define a ordenação do conjunto de aulas.

O Modelo Linear Generalizado para o caso do AGC não possui boa aderência, sendo que o valor de  $R^2$  ajustado do modelo é de apenas 65,35%, fato explicado parcialmente por uma aparente disruptura nos resultados obtidos para algumas populações. A Figura 34 ilustra os resultados obtidos por um *scatterplot*. Nota-se que para o Peso 1 e  $Ordem(A) = 1$ , tem-se um comportamento peculiar do GAP para  $Pop$  entre 2000 e 3000.

Continuando as análises, com intuito de obter os parâmetros mais robustos do AGC, é evidenciado o impacto na escolha da forma de amostragem para o problema. A amostragem com ordenação por demanda possui indício nos dados de ser melhor que a amostragem com ordenação aleatória. Finalmente a análise de Desejabilidade Composta indica que  $Ordem(A)$  por demanda e  $Pop = 3700$  serão melhores parâmetros para o AGC para este experimento, outros resultados são exibidos na Tabela 9.

O Algoritmo Genético proposto possui três parâmetros:  $T_C$ ,  $T_M$  e  $Pop$ , sendo a taxa de *Crossover* (Recombinação), taxa de mutação e tamanho da população, respectivamente. O modelo linear generalizado para este problema obteve  $R^2$  ajustado de 87,26%, possuindo assim uma boa aderência aos dados. Diferente do GRASP<sub>T</sub>, o Algoritmo Genético é bem influenciado pela escolha de seus parâmetros, como ilustra a Figura 35, porém ainda permanece com alto impacto devido à escolha dos pesos.

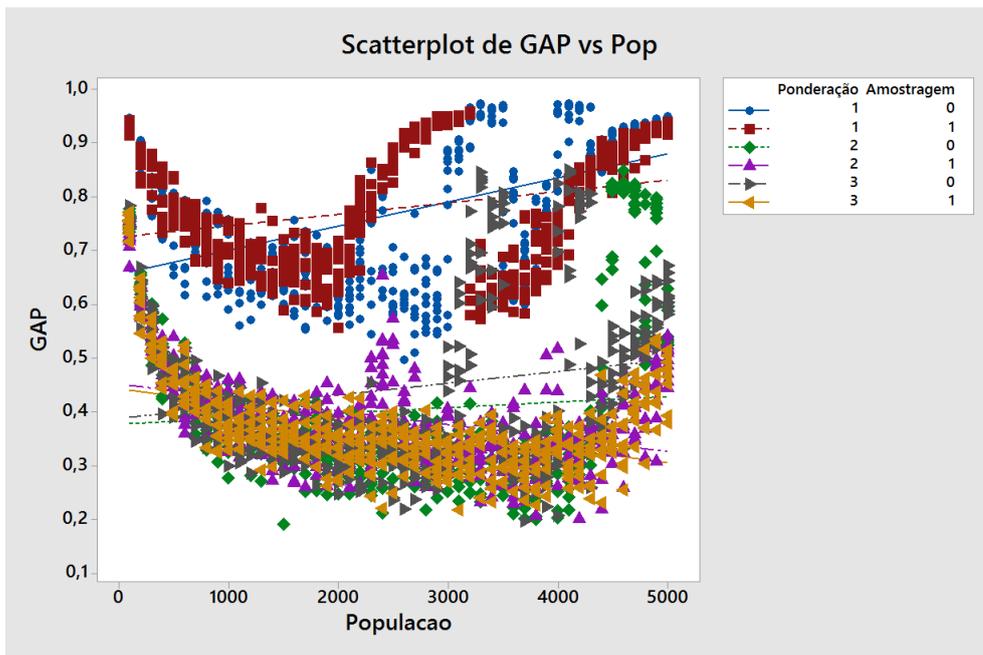


Figura 34: *Scatterplot* do ensaio do AGC

Tabela 9: Composite Desirability para as combinações de parâmetros do AGC

Peso	Amostragem	Pop	GAP Ajustado	Desejabilidade Composta
2	1	3700	0,273951	0,893429
2	1	2000	0,283168	0,881663
3	1	3700	0,287226	0,876484
2	1	3800	0,290542	0,872250
2	1	3600	0,293591	0,868357
3	1	2000	0,296443	0,864718
2	1	1900	0,297668	0,863153
2	0	3700	0,299538	0,860767
2	1	1800	0,299736	0,860514
2	1	2100	0,300231	0,859882

A Figura 35 ilustra também que a taxa de mutação  $T_M$  influencia com mais intensidade a qualidade das soluções obtidas do que a taxa de crossover  $T_C$ . O tamanho da população possui um impacto intermediário entre os dois. Esta diferença entre o impacto dos parâmetros pode ser observada nas tabelas de agrupamento do teste de Turkey pareado, Tabelas 10 e 11, respectivamente.

As Tabelas 10 e 11 ilustram os agrupamentos dos níveis para a taxa de *crossover* e mutação respectivamente. Não há indícios nos dados de que haja diferença significativa entre os níveis que estão em um mesmo agrupamento. Nota-se que existem somente 5 agrupamentos para a taxa de *crossover* e que para os valores 1,0, 0,7, 0,8 e 0,9 pertencem ao mesmo agrupamento. Portanto, a escolha de  $T_C = 1,0$  como melhor parâmetro

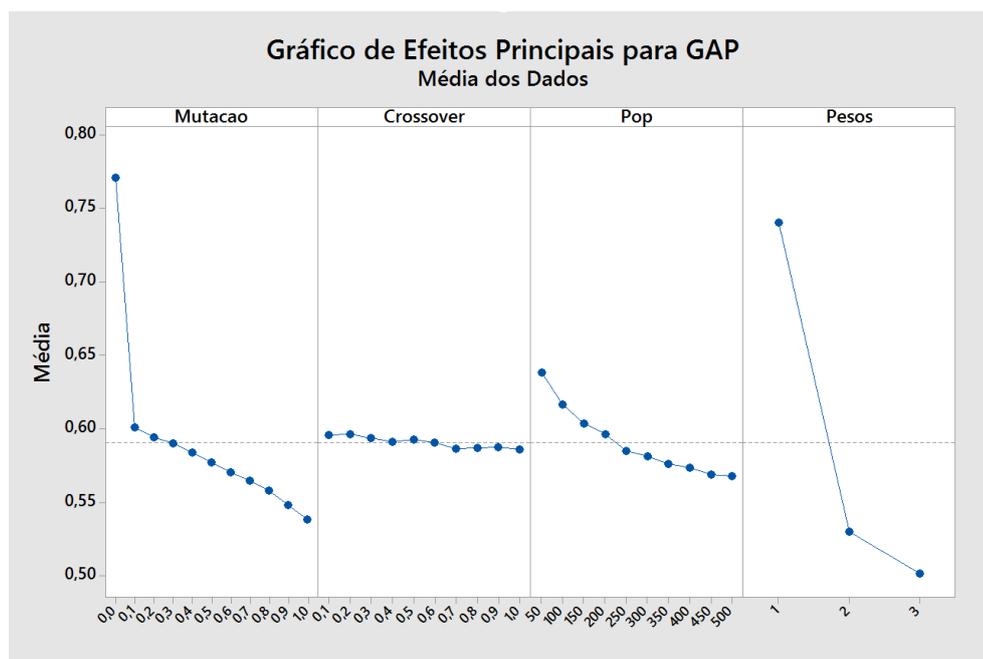


Figura 35: Gráfico de Efeitos Principais para o Algoritmo Genético

Tabela 10: Teste de Turkey pareado para Taxa de *Crossover*

Crossover	N amostral	Média	Agrupamento				
			A	B	C	D	E
0,2	3300	0,596707	●				
0,1	3300	0,595912	●	●			
0,3	3300	0,593801	●	●	●		
0,5	3300	0,592635		●	●		
0,4	3300	0,591219			●	●	
0,6	3300	0,590813			●	●	
0,9	3300	0,587777				●	●
0,8	3300	0,587085					●
0,7	3300	0,586543					●
1,0	3300	0,586012					●

pode não ser tão apropriada quanto  $T_C = 0,7$ , por exemplo. Entretanto, para a taxa de mutação, nota-se que dentre os 11 possíveis níveis foram feitos 10 agrupamentos, indicando evidência nos dados que a escolha do parâmetro  $T_M$  altera significativamente a performance do método.

Calculando portanto o Desejabilidade Composta para o Algoritmo Genético, obtém-se como configuração mais indicada de parâmetros:  $T_C = 1,0$ ,  $T_M = 1,0$  e  $Pop = 500$ . Destaca-se porém que a escolha de  $T_C = 1,0$  não é necessariamente a melhor e pode se esperar que  $T_C = 0,8$  tenha um desempenho similar, em contraste com a escolha dos parâmetros de  $T_M$  e  $Pop$ .

Tabela 11: Teste de Turkey pareado para Taxa de Mutação

Mutação	N amostral	Média	Agrupamento											
			A	B	C	D	E	F	G	H	I	J		
0,0	3000	0,771053	•											
0,1	3000	0,601168		•										
0,2	3000	0,594245			•									
0,3	3000	0,590473			•									
0,4	3000	0,584148				•								
0,5	3000	0,577420					•							
0,6	3000	0,570757						•						
0,7	3000	0,565096							•					
0,8	3000	0,558324								•				
0,9	3000	0,548292									•			
1,0	3000	0,538380										•		

A Busca Tabu, similarmente ao  $GRASP_T$  apresenta um baixo impacto da escolha do parâmetro *Tabu* na performance do método. Portanto o tamanho da Lista Tabu irá ter um efeito menor do que a escolha dos pesos, de responsabilidade dos decisores. Ilustrando este comportamento, pode-se perceber na comparação entre as Figuras 33 e 36 que para o parâmetro *Tabu* da BT e o parâmetro *Tabu* do  $GRASP_T$ , possuem comportamento parecido.

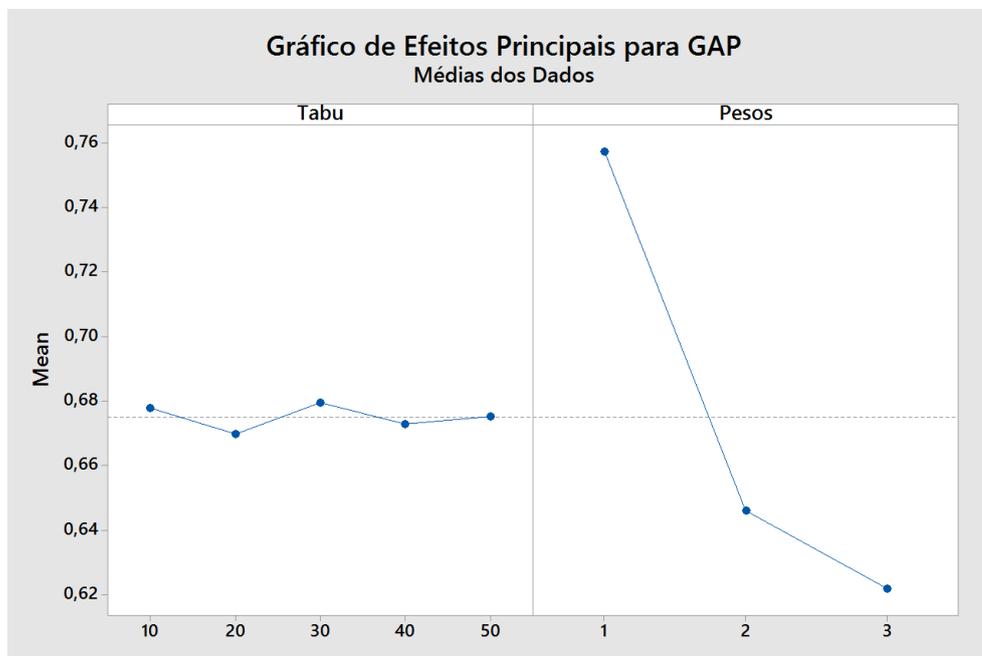


Figura 36: Gráfico de Efeitos Principais para a Busca Tabu

Defrontando-se novamente com a questão do baixo impacto da escolha do parâmetro que foi visto em  $GRASP_T$ , foi realizado então para um teste pareado sobre as médias. A Figura 37 exhibe os resultados dos testes de Turkey pareados com intervalos de confiança

simultâneos de 95%. Como todos os intervalos contém zero, não é possível diferenciar os impactos da escolha do parâmetro e qualquer valor optado pertencerá a um mesmo agrupamento. Portanto a escolha para o parâmetro neste caso é arbitrária. A fim de manter a coerência será adotado como parâmetro *Tabu* da Busca Tabu o mesmo valor utilizado no  $GRASP_T$ , i.e.,  $Tabu = 40$ .

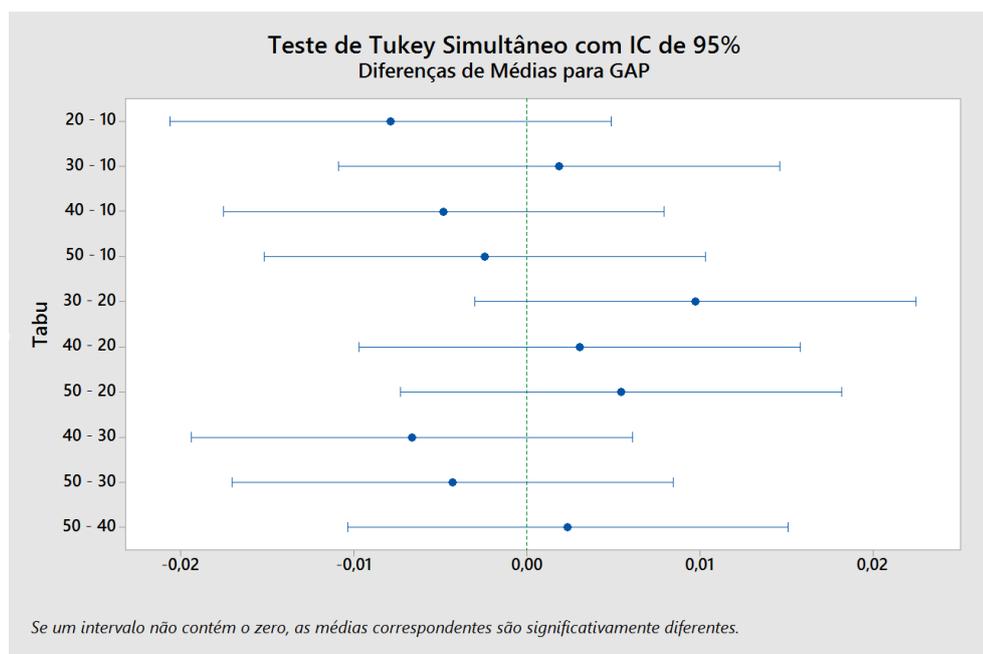


Figura 37: Teste de Turkey Pareado para *Tabu* na metaheurística Busca Tabu

Finalmente, a Tabela 12 indica os valores utilizados de parâmetros para cada um dos métodos implementados, resumindo portanto os resultados obtidos com os testes de sensibilidade de parâmetros.

Tabela 12: Parâmetros escolhidos para cada um dos métodos mono-objetivo

Método	Parâmetros Escolhidos
AG	$T_C = 1, 0, T_M = 1, 0, Pop = 500$
AGC	$Ordem(A) = 1, Pop = 3700$
GRASP	$L = 4, N = 10$
$GRASP_T$	$L = 2, N = 10, Tabu = 40$
Busca Tabu	$Tabu = 40$
Busca Local	Não-Paramétrica

## 6.5 Desempenho das Abordagens Mono-objetivo

Após a realização dos testes de sensibilidade de parâmetros, Seção 6.4, os parâmetros dos métodos são fixados segundo a Tabela 12. O experimento para testar o desempenho

dos métodos é composto de ensaios de 30 execuções de 5 minutos para cada algoritmo, instância e peso. Cada execução  $k$ , do método  $m$ , irá obter um valor de função objetivo  $F_m^k(i, p)$  para a instância  $i$ , com o peso  $p$ .

$$GAP_m^k(i, p) \doteq \frac{F_m^k(i, p) - \hat{F}(i, p)}{F_m^k(i, p)} \quad (6.3)$$

Para cada par instância peso, é utilizado o limitante inferior  $\hat{F}(i, p)$  da Tabela 5, para obter o GAP de otimalidade, dado pela expressão (6.3). A Figura 38 apresenta um *boxplot* do experimento realizado, em que a variação do GAP por cada método é apresentada.

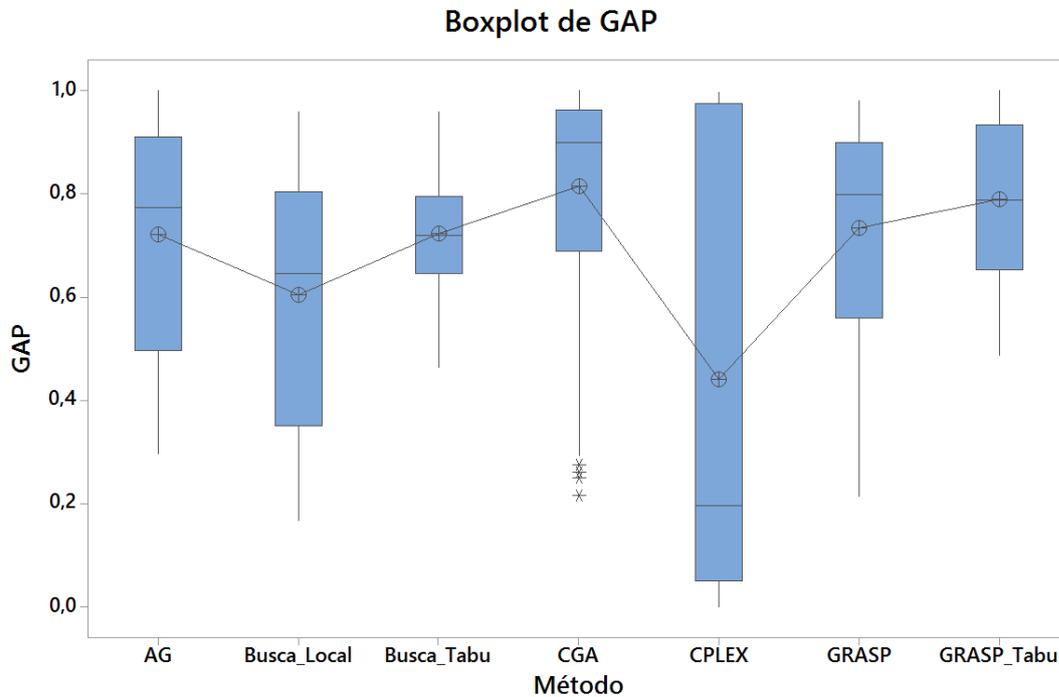
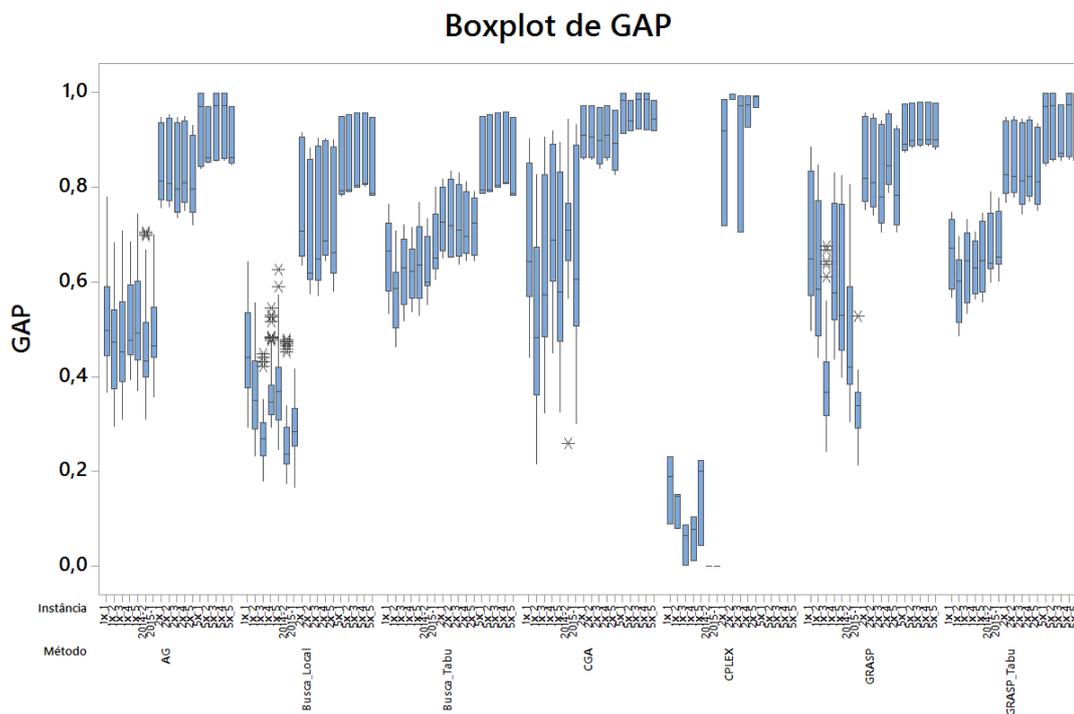


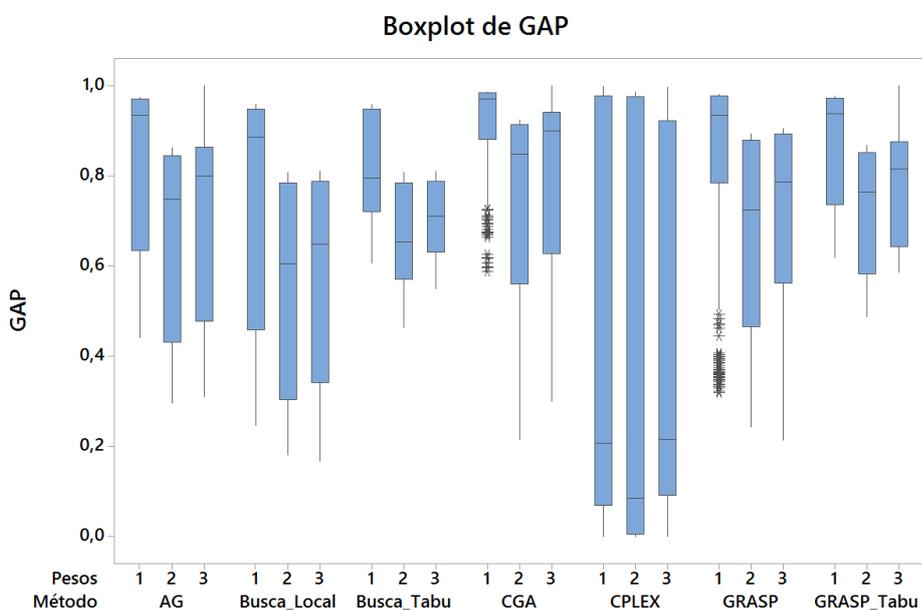
Figura 38: *Boxplot* do experimento de desempenho dos métodos mono-objetivo

É possível notar que a Busca Tabu e o AGC são aparentemente os métodos com a pior performance. O AGC possui vários *outliers* por volta dos 30% de GAP de otimalidade. A fim de entender os *outliers*, a Figura 39, apresenta a subdivisão do boxplot em outras variáveis, identificando a variação da performance dos métodos conforme o peso ou a instância a ser resolvida. A Figura 39(a), é o *boxplot* subdividido pelos métodos e instâncias, enquanto a Figura 39(b) ilustra a subdivisão por método e peso.

São aplicados a estes resultados um teste de Games-Howell, com intervalos de confiança simultâneos de 95%, pois com o tamanho de amostra pequena não foi assumido variância amostral idêntica. Este teste, verifica se as diferenças médias de dois métodos,



(a) Método e Instância

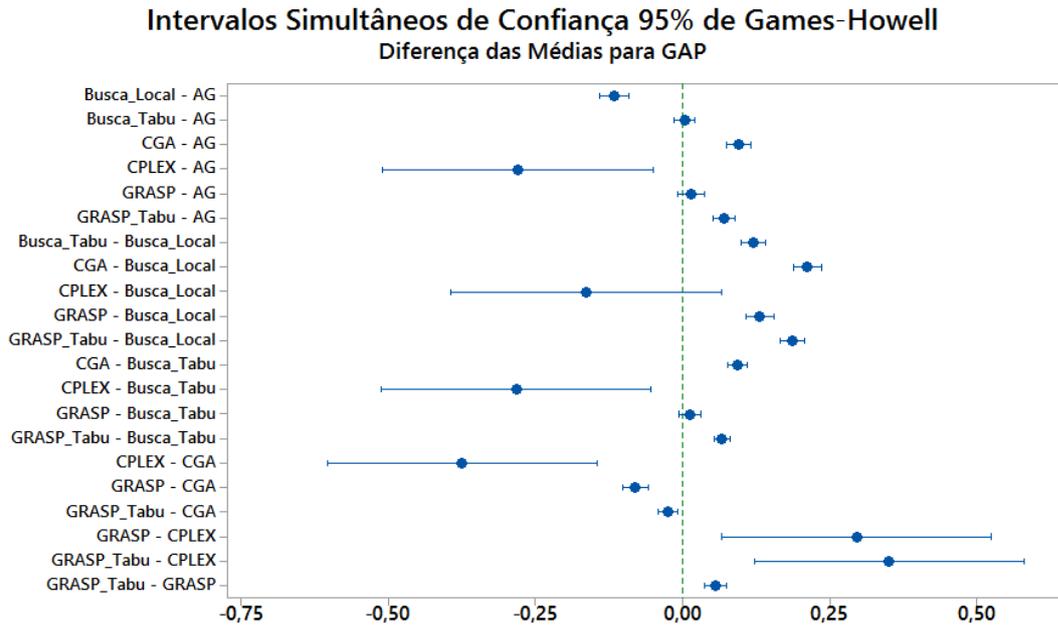


(b) Método e Peso

Figura 39: *Boxplot* do desempenho dos métodos subdividido

tomados dois a dois, possuem o zero. Caso o intervalo de confiança não contenha o zero, as médias dos métodos testados são significativamente diferentes. Caso contrário, os métodos são estatisticamente similares. A Figura 40 ilustra o teste descrito.

Pelo resultado do teste de Games-Howell, apresentado na Figura 40, o CPLEX apre-



Se um intervalo não contém o zero, as médias correspondentes são significativamente diferentes.

Figura 40: Teste pareado de Games-Howell para o desempenho dos métodos mono-objetivo

sendo o melhor resultado com significância estatística, que todos os outros métodos, exceto a busca local. Como este teste é baseado nas médias, ele é insensível à *outliers* e amostras pequenas, sendo estes dois problemas apresentados pelos dados.

Para contornar o viés da média e os problemas com baixa amostragem, é proposto também um teste de Dolan-Moré (DOLAN; MORÉ, 2002). Este é um típico teste de performance de algoritmos sobre várias instâncias, com a propriedade de lidar com resultados faltantes e *outliers*. O teste também foi realizado utilizando o *GAP* mediano dos experimentos para cada instância, trocando o viés da média pelo viés da mediana.

O teste de performance de Dolan-Moré avalia um conjunto de métodos ( $\mathcal{S}$ ) com relação a um conjunto de instâncias ( $\mathcal{P}$ ). Atribuindo uma métrica de qualidade  $t_{p,s}$  para cada método  $s \in \mathcal{S}$  e instância  $p \in \mathcal{P}$ , que deverá ser minimizada. A partir desta métrica define-se o raio de performance, dado pela expressão (6.4), que fornece informações sobre a performance do método  $s$  em uma instância  $p$  com relação à melhor performance de todos os métodos.

$$r_{p,s} = \frac{t_{p,s}}{\min_{p \in \mathcal{P}} \{t_{p,s}\}} \quad (6.4)$$

Define-se ainda um valor  $r_{max} \geq r_{p,s}$  (a escolha de  $r_{max}$  não afeta o teste) e toda instância  $s$  que o método  $p$  não for capaz de resolver, faz-se  $r_{p,s} = r_{max}$ . Pelo raio de performance é definido  $\rho_s(\tau)$ , pela expressão (6.5), a probabilidade de que o raio de performance do método  $s$  esteja dentro de um fator  $\tau$  do melhor raio de performance. Esta será portanto uma distribuição de probabilidade acumulada para o raio de performance, será chamado de perfil de desempenho do método  $m$ .

$$\rho_s(\tau) = \frac{1}{|\mathcal{P}|} |\{p \in \mathcal{P} | r_{p,s} \leq \tau\}| \quad (6.5)$$

Portanto, com o teste de Dolan-Moré são buscados os métodos que possuem distribuição acumulada igual a um com o menor fator  $\tau$ . Caso um método  $s \in \mathcal{S}$  não seja capaz de resolver uma instância  $p \in \mathcal{P}$ , a distribuição acumulada será um somente no limite de  $\tau \rightarrow \infty$ . Portanto para uma pequena variação de  $\tau$ , o perfil de desempenho irá chegar, no máximo, ao percentual de instâncias resolvidas.

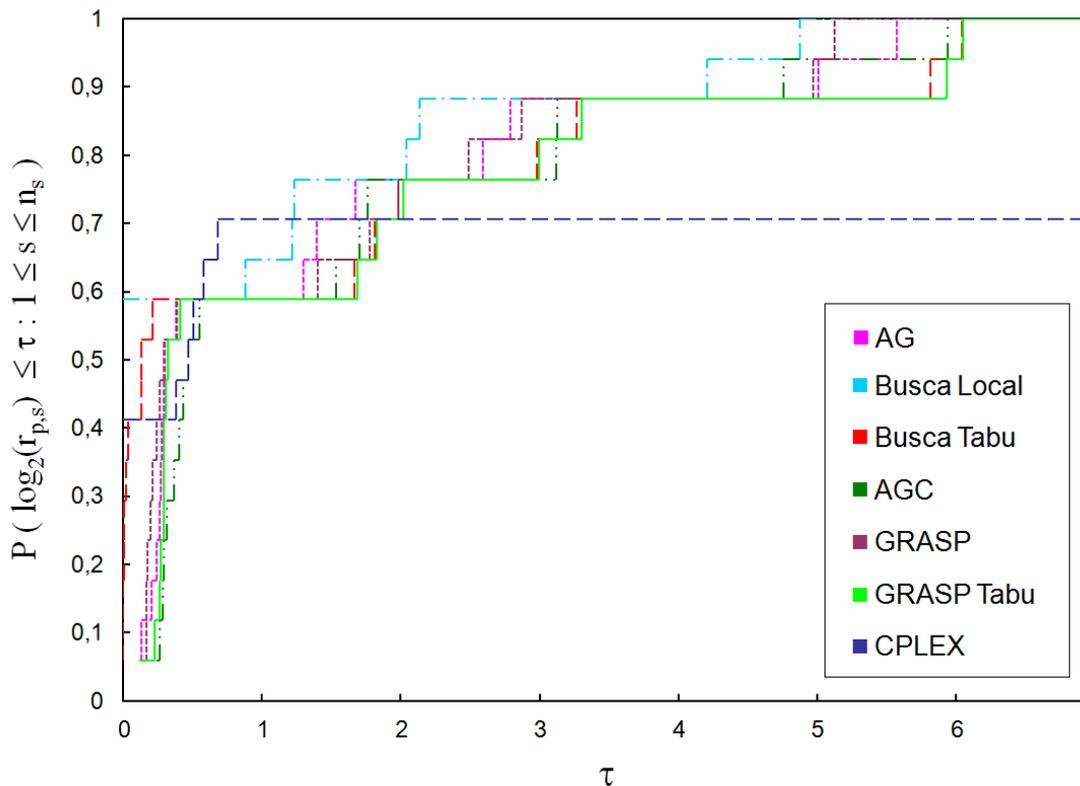


Figura 41: Teste de desempenho de Dolan-Moré

O resultado do teste de performance de Dolan-Moré é apresentado na Figura 41. É possível notar que a performance do CPLEX é superior aos outros métodos para os primeiros perfis de desempenho, porém como o CPLEX não é capaz de resolver as instâncias  $5x$ , o seu perfil de desempenho será 70%. A busca local e o GA mostram bons desempe-

nhos para um grande número de instâncias, porém a busca local domina o AG. Portanto, a busca local é a melhor forma de resolução do PAAS, dentre todas as testadas nesta dissertação.

## 6.6 Desempenho dos Métodos Multi-objetivo

O algoritmo considerado foi o ABT, Capítulo 5, com os 3 tipos de busca: AGC modificado (Algoritmo 5.2), busca local modificada (Algoritmo A.1) e busca tabu modificada (Algoritmo A.2). Os parâmetros  $TT$  de tamanho máximo das tabelas e  $t_b$  de tempo máximo de busca variam conforme ilustra a Tabela 13.

Tabela 13: Parâmetros utilizados para o ABT

Parâmetro	Valores
$TT$ Tamanho Máximo das Tabelas	{200, 400, 600, 800, 1000}
$t_b$ Tempo Máximo de Busca	{5s, 15s, 25s, 35s, 45s}

Para cada execução do ABT com uma dada busca, existem 25 possibilidades de configurações de parâmetros. Cada uma destas combinações é executada uma vez, com  $t_{max} = 900s$ , e as direções de interesse dadas pela Tabela 4, na página 61. As instâncias utilizadas para testes são as 11 das 17 instâncias descritas em 6.1, sendo as 2 reais do ICMC, 4 instâncias **1x**, 2 instâncias **2x** e 3 instâncias **5x**.

Define-se  $\mathbb{Y}^i$ , o conjunto solução obtido pelo ABT na instância  $i$ , para todas as buscas e combinações de parâmetros. Seja  $z^{max,i}$  dado pela expressão (6.6), define-se  $G(i) = \{\frac{1}{z_1^{max,i}}, \dots, \frac{1}{z_d^{max,i}}\}$ . A aplicação  $T^i(Y) = Y \cdot G(i)$ , para  $Y \in \mathbb{Y}^i$  leva soluções no espaço  $Z$ , para o espaço  $\bar{Z} \subseteq [0, 1]^d$ , i.e.,  $T : Z \rightarrow \bar{Z}$ .

$$z_j^{max,i} = \max_{Y \in \mathbb{Y}^i} \{f_j(Y)\} \quad (6.6)$$

Para o cálculo do hipervolume dos conjuntos solução, deve se definir um ponto de referência único para cada instância. No espaço de objetivos normalizado  $\bar{Z}$ , o ponto de referência será definido por  $z^{Ref,i} \doteq \vec{\mathbf{1}}$ . Para o cálculo do hipervolume, foi utilizado o algoritmo proposto por Fonseca et al. (2006). O resultado dos experimentos são ilustrados pelo *boxplot*, na Figura 42, em que o hipervolume para cada busca é exibido.

Nota-se que o AGC possui uma média e mediana baixas para o problema, porém os melhores resultados obtidos também são por este método, caracterizado pelo grande número de *outliers*. Para melhor entender este comportamento, o *boxplot* do AGC foi

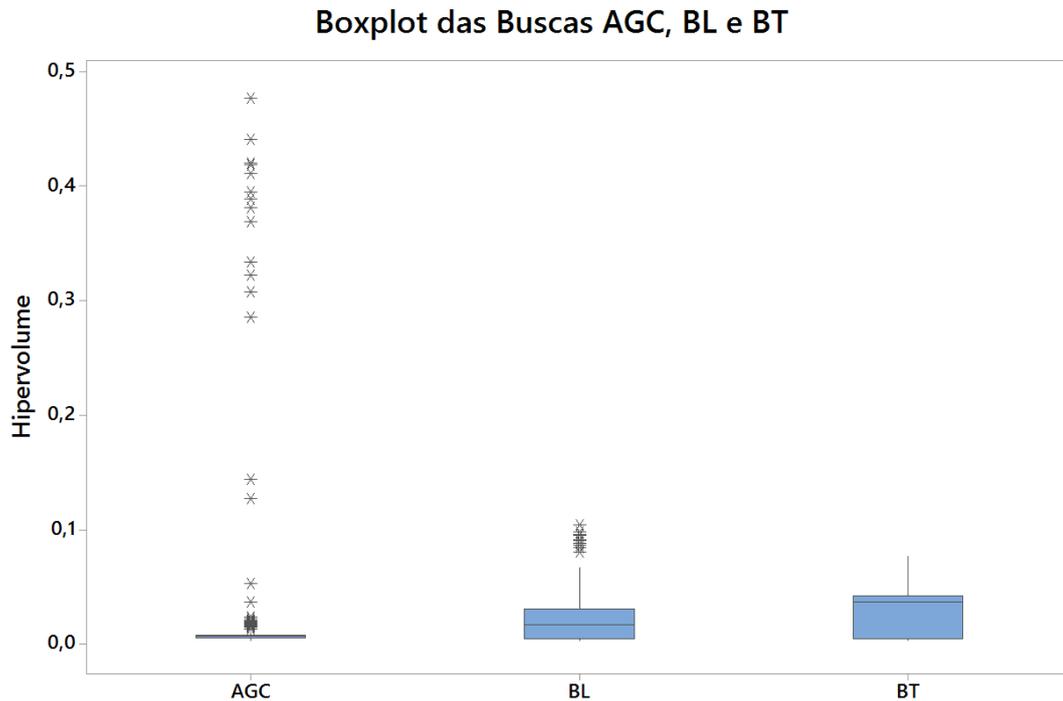


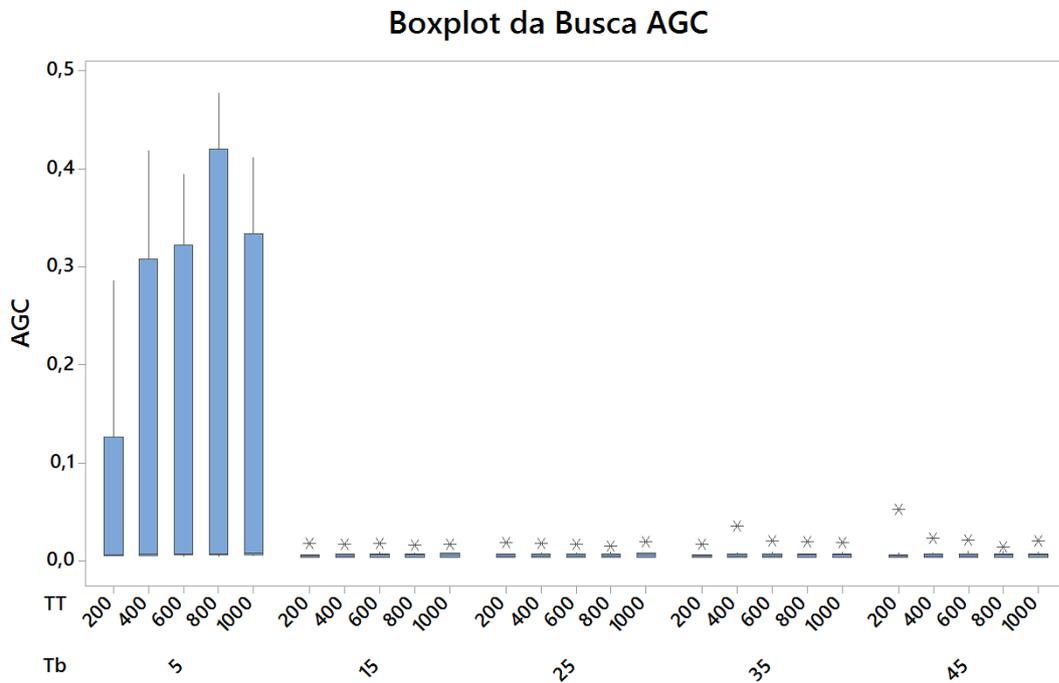
Figura 42: Boxplot do hipervolume para as Buscas AGC, BL e BT

dividido em categorias, como ilustra a Figura 43.

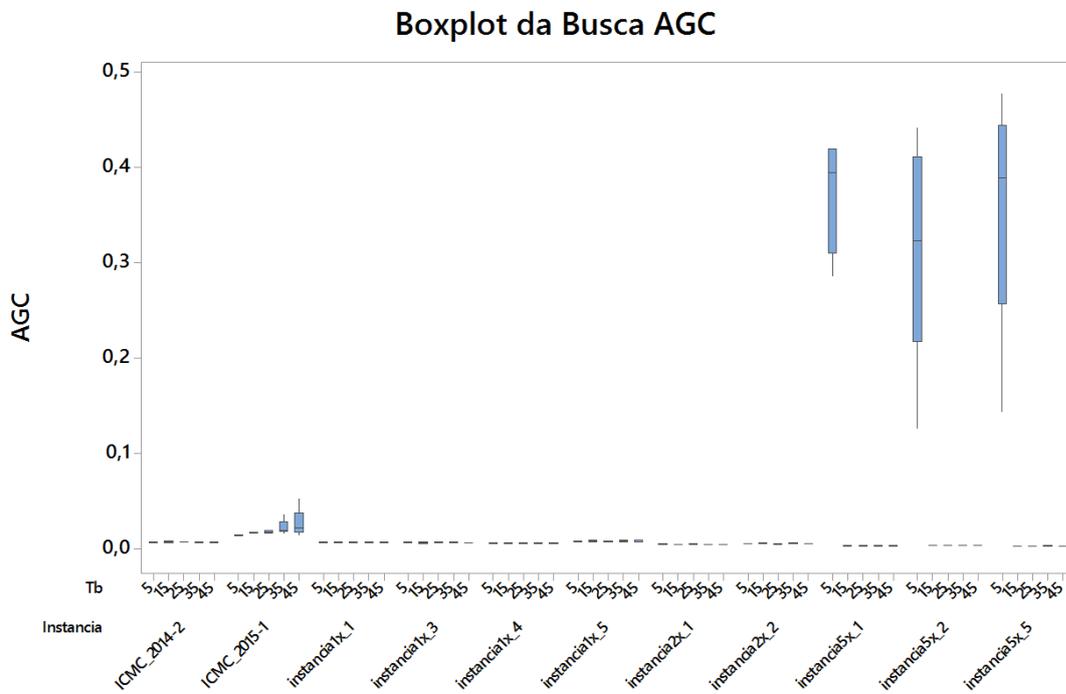
A Figura 43(a) ilustra a divisão por tamanho de tabelas  $TT$  e tempo máximo de busca  $t_b$ , enquanto a Figura 43(b) ilustra a divisão por tempo máximo de busca  $t_b$  e instância. Nota-se que os *outliers* desaparecem nestas figuras, ou seja, o hipervolume obtido varia bastante pelas instâncias e tempos de busca, porém não há *outliers* dentro destas.

Inicialmente é feito um teste t pareado com as 3 buscas, a fim de estabelecer uma relação entre as médias. A Figura 44 ilustra o teste t realizado com intervalo de confiança de 95%, com hipótese nula de que a média de uma busca é estritamente maior que de outra. Caso a média do intervalo de confiança seja maior que zero, existe evidência nos dados de que a média do primeiro supera a média do segundo, como é o caso da BT com a BL na Figura 44(c). Não se pode afirmar nada quanto à média do AGC com relação ao BL e ao BT.

A busca local, que foi o melhor método mono-objetivo implementado (como visto na Seção 6.5), é o pior tipo de busca para o ABT. Ao passo que o AGC, que não possui uma performance mono-objetivo de destaque, mostrou-se extremamente superior às outras buscas para instâncias grandes, pois apresenta hipervolumes de até 45% para estas instâncias enquanto a média das outras buscas não passa de 10%. Para instâncias menores, entretanto, o BT se sobressai.

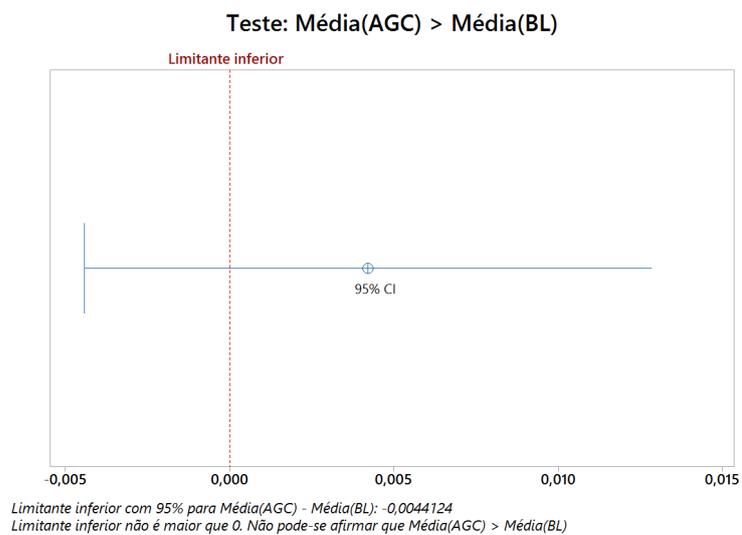


(a) Tempo de Busca e Tamanho de Tabela

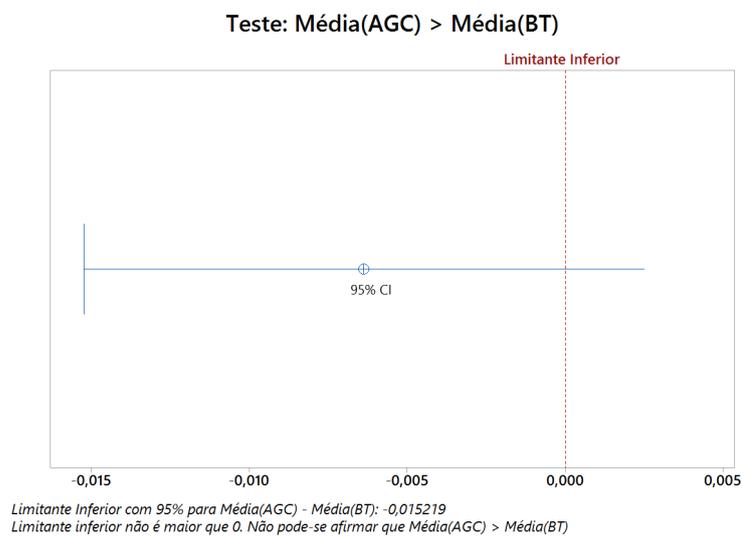


(b) Tempo de Busca e Instâncias

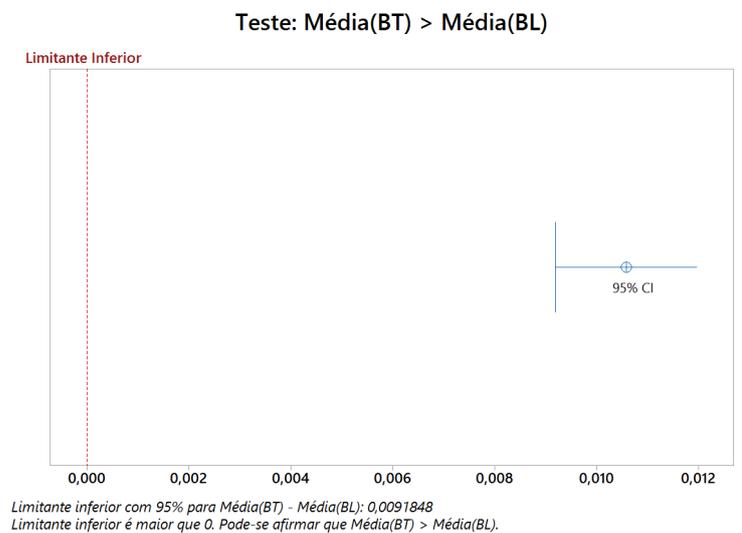
Figura 43: *Boxplot* da Busca AGC dividida em categorias



(a) AGC e BL



(b) AGC e BT



(c) BT e BL

Figura 44: Teste t pareado para a média dos hipervolumes das buscas



---

---

## CAPÍTULO 7

---

### CONCLUSÕES E PERSPECTIVAS

Na presente Dissertação foi tratado o Problema de Alocação de Aulas à Salas (PAAS), tomando como base um caso de estudo no Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP) no campus São Carlos. Foi estabelecida uma relação entre as métricas consideradas para este trabalho e as já utilizadas na literatura que trata o PAAS.

Para este caso de estudo, foi proposto um modelo matemático de programação inteira, abordado por três frentes distintas: resolução exata do modelo, metaheurísticas mono-objetivo e uma metaheurística multi-objetivo. Os métodos mono-objetivo propostos passaram por um ajuste de parâmetros, feito por meio de um experimento estatístico.

A resolução pelo método exato foi utilizada para comparação com as metaheurísticas, com o tempo de execução igual. Também foi executada por uma hora, a fim de fornecer um limitante inferior, possibilitando normalizar o resultado dos experimentos, por meio do GAP de otimalidade.

Foram realizados experimentos estatísticos com os métodos mono-objetivo para determinar qual dos métodos propostos resolveria o problema com mais eficiência. Foi verificado que a busca local proposta possui performance superior ao CPLEX, com o mesmo tempo de execução, verificada por testes de hipótese e um método *benchmark* de avaliação de performance.

Para aferir qualidade à solução multi-objetivo, é utilizado o indicador de hipervolume.

Para o método multi-objetivo proposto, foram executados alguns ensaios estatísticos, a fim de determinar a melhor configuração de parâmetros, com base no hipervolume. Foi obtido um resultado com o multi-objetivo, complementar ao mono-objetivo, pois métodos mono-objetivo com baixo desempenho, mostraram-se promissores na estratégia multi-objetivo.

Trabalhos futuros devem incluir investigações sobre de novos métodos multi-objetivo, já que a aplicação da abordagem mostrou-se promissora. Investigações sobre redução de dimensão, por exemplo, análise de componentes principais, ou visualização para o espaço de objetivos, se apresentam como possibilidades interessantes.

Sobre as técnicas mono-objetivo, a boa performance do AG indica que investigações sobre novos operadores genéticos podem melhorar sua performance. Considerando ainda que a busca local foi a melhor metaheurística investigada neste trabalho, a utilização de um algoritmo memético mostra-se promissora.

---

## REFERÊNCIAS BIBLIOGRÁFICAS

BEHESHTI, Z.; SHAMSUDDIN, S. M. Non-parametric particle swarm optimization for global optimization. *Applied Soft Computing*, Elsevier B.V., v. 28, p. 345–359, 2015. ISSN 15684946.

CARTER, M. W.; TOVEY, C. A. When is Classroom assignment problem hard? *Operations Research*, v. 40, n. 1-supplement-1, p. S28–S39, 1992.

CIRINO, R. B. Z.; COSTA, A. M.; SANTOS, M. O. Um Modelo Matemático para a Resolução do Problema de Alocação de Salas no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo. *Anais do XLV Simpósio Brasileiro de Pesquisa Operacional*, Natal, RN, p. 2555–2565, 2013.

CIRINO, R. B. Z.; SANTOS, M. O.; DELBEM, A. C. B. Aplicação da Metaheurística AGC para o Problema de Alocação de Aulas a Salas. *XLVII SBPO - Simpósio Brasileiro de Pesquisa Operacional*, Porto de Galinhas, PE, p. 1745–1755, 2015.

COELLO, C. a. C.; LAMONT, G. B.; VELDHUIZEN, D. a. V. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. 2<sup>ª</sup> ed. New York, NY: Springer, 2002. 822 p. ISBN 9780387310299.

CONSTANTINO, A. A.; Marcondes Filho, W.; LANDA-SILVA, D. Iterated heuristic algorithms for the classroom assignment problem. *Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling - PATAT, Belfast*, p. 152–166, 2010.

DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. *Mathematical Programming, Series B*, v. 91, n. 2, p. 201–213, 2002. ISSN 00255610.

DYER, J. S.; MULVEY, J. M. An Integrated Optimization/Information System for Academic Departmental Planning. *Management Science*, v. 22, n. 12, p. 1332–1341, 1976. ISSN 0025-1909.

- EIBEN, A.; SMIT, S. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, Elsevier B.V., v. 1, n. 1, p. 19–31, 2011. ISSN 22106502.
- ELLOUMI, A.; KAMOUN, H.; JARBOUI, B.; DAMMAK, A. The classroom assignment problem: Complexity, size reduction and heuristics. *Applied Soft Computing*, Elsevier B.V., v. 14, p. 677–686, 2014. ISSN 15684946.
- FEO, T. a.; RESENDE, M. G. C. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, v. 6, n. 2, p. 109–133, 1995. ISSN 09255001.
- FERLAND, J. A.; ROY, S. Timetabling problem for university as assignment of activities to resources. *Computers & Operations Research*, v. 12, n. 2, p. 207–218, jan 1985. ISSN 03050548.
- FITRIANTO, A.; MIDI, H. Multi-Response Optimization via Desirability Function for the Black Liquor DATA. *Journal of Science and Technology*, n. II, p. 91–102, 2011.
- FONSECA, C.; PAQUETE, L.; LOPEZ-IBANEZ, M. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. *2006 IEEE International Conference on Evolutionary Computation*, p. 1157–1163, 2006.
- FONSECA, V. G.; FONSECA, C. M.; HALL, A. O. Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function. *Lecture Notes in Computer Science/Evolutionary Multi-Criterion Optimization*, v. 1993, p. 213–225, 2001. ISSN 16113349.
- FUTUYAMA, D. J. *Biologia Evolutiva*. 2. ed. Ribeirão Preto: Sociedade Brasileira de Genética/CNPq, 1992. 646 p.
- GLOVER, F. Paths for Integer Programming and links to artificial intelligence. *Computers and Operations Research*, v. 13, n. 5, p. 533–549, 1986.
- HARIK, G. R.; LOBO, F. G.; GOLDBERG, D. E. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, v. 3, n. 4, p. 287–297, 1999. ISSN 1089778X.
- HOLLAND, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Viii. Oxford, England: MIT Press Cambridge, 1975. 183 p.
- HUNDT, R. Loop Recognition in C ++ / Java / Go / Scala. *Proceedings of Scala Days*, Stanford, v. 1, n. 1, p. 38–47, 2011.
- KRIPKA, R. M. L.; KRIPKA, M. Simulated Annealing Aplicado Na Otimização Da Alocação de Salas em Instituição de Ensino Superior. *Mecánica Computacional*, XXIX, p. 9317–9325, 2010.
- LAGUNA, M.; VELARDE, J. L. G. A Search Heuristic for Just-in-Time Scheduling in Parallel Machines. *Journal of Intelligent Manufacturing*, v. 2, n. 4, p. 253–260, 1991.
- LENSTRA, J. K. *Local Search in Combinatorial Optimization*. Princeton: Princeton University Press, 1997. 512 p. ISBN 9780691115221.

LOPES, M. C.; SCHOEFFEL, P. S. Um método de Alocação para o problema de Reservas de Sala de Aula. *II Congresso Brasileiro de Computação*, 2002.

MARTINEZ-ALFARO, H.; FLORES-TERAN, G. Solving the classroom assignment problem with simulated annealing. In: *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*. San Diego: IEEE, 1998. v. 4, p. 3703–3708. ISBN 0-7803-4778-1.

MISHRA, K. K.; HARIT, S. A Fast Algorithm for Finding the Non Dominated Set in Multi objective Optimization. *International Journal of Computer Applications*, v. 1, n. 25, p. 35–39, 2010.

MOONEY, E. L.; RARDIN, R. L.; PARMENTER, W. J. Large scale classroom scheduling. *IIE transactions*, v. 28, n. 5, p. 369–378, 1996.

MULVEY, J. M.; SCIENCE, A.; SEPTEMBER, R.; COMPANY, N.-h. P. A classroom/time assignment model. *European Journal of Operational Research*, v. 9, n. 1, p. 64–70, 1982. ISSN 0377-2217.

NASCIMENTO, a. S.; SAMPAIO, R. M.; ALVARENGA, G. B. Uma aplicação de simulated annealing para o problema de alocação de salas. *INFOCOMP Journal*, p. 59–66, 2005.

PEIRÓ, J.; CORBERÁN, Á.; MARTÍ, R. GRASP for the uncapacitated r-allocation p-hub median problem. *Computers and Operations Research*, Elsevier, v. 43, n. 1, p. 50–60, 2014. ISSN 03050548.

PHILLIPS, A. E.; WATERER, H.; EHRGOTT, M.; RYAN, D. M. Integer programming methods for large-scale practical classroom assignment problems. *Computers & Operations Research*, Elsevier, v. 53, p. 42–53, 2015. ISSN 03050548.

PILLAY, N. A survey of school timetabling research. *Annals of Operations Research*, v. 218, n. 1, p. 261–293, 2014. ISSN 0254-5330.

PRADO, A. S.; SOUZA, S. R. de. Problema de alocação de salas em cursos universitários: um estudo de caso. *Anais do XLVI Simpósio Brasileiro de Pesquisa Operacional*, Salvador, p. 2054–2065, 2014.

SANTOS, a. C.; DELBEM, a. C. B.; LONDON, J. B. a.; BRETAS, N. G. Node-depth encoding and multiobjective evolutionary algorithm applied to large-scale distribution system reconfiguration. *IEEE Transactions on Power Systems*, v. 25, n. 3, p. 1254–1265, 2010. ISSN 08858950.

SANTOS, M. O.; MASSAGO, S.; ALMADA-LOBO, B. Infeasibility handling in genetic algorithm using nested domains for production planning. *Computers and Operations Research*, v. 37, n. 6, p. 1113–1122, 2010. ISSN 03050548.

SASTRY, K.; GOLDBERG, D.; KENDALL, G. Genetic Algorithms. In: *Search Methodologies*. New York, NY: Springer US, 2005. cap. 4, p. 97–125. ISBN 978-1-4614-6939-1.

- SAVINIEC, L.; CONSTANTINO, A. A.; ROMÃO, W. Vns based algorithms to the high school timetabling problem. *Anais do XLV Simpósio Brasileiro de Pesquisa Operacional*, Natal, p. 845–856, 2013.
- SILVA, D. J. da; SILVA, G. C. da. Heurísticas baseadas no algoritmo de coloração de grafos para o Problema de alocação de salas em uma instituição de ensino superior. *Anais do XLII SBPO, Bento Gonçalves*, p. 2839–2849, 2009.
- SILVA, G. C. da; BAHIENSE, L.; Satoru Ochi, L.; BOAVENTURA-NETTO, P. O. The dynamic space allocation problem: Applying hybrid GRASP and Tabu search metaheuristics. *Computers & Operations Research*, Elsevier, v. 39, n. 3, p. 671–677, 2012. ISSN 03050548.
- SOUZA, M. J. F.; MACULAN, N.; OCHI, L. S. A GRASP-Tabu search algorithm for solving school timetabling problems. In: *Metaheuristics: Computer decision-making*. Boston, MA: Springer US, 2004, (Applied Optimization). p. 627–657. ISBN 978-1-4419-5403-9.
- SOUZA, M. J. F.; MARTINS, A. X.; ARAÚJO, C. R. D. Experiências com Simulated Annealing e Busca Tabu na Resolução do Problema de Alocação de Salas. *Anais do XXXIV Simpósio Brasileiro de Pesquisa Operacional*, Rio de Janeiro, p. 1100–1110, 2002.
- SOUZA, M. J. F.; MARTINS, A. X.; ARAÚJO, C. R. D.; AZEVEDO, F. W. C. Métodos de Pesquisa em Vizinhança Variável aplicados ao Problema de Alocação de Salas. *XXII Encontro Nacional de Engenharia de Produção*, p. 1–8, 2002.
- SOUZA, P. L. B. D.; SCARPIN, C. T. Aplicação De Algoritmos Genéticos E Métodos Evolucionários Na Resolução Do Problema De Alocação De Turmas: Caso Ufpr. In: *Anais do XVII Simpósio de Pesquisa Operacional e Logística da Marinha*. São Paulo: Editora Edgard Blücher, 2014. p. 625–636.
- SUBRAMANIAN, A.; MEDEIROS, J. M. F.; FORMIGA, L. D. A.; SOUZA, M. J. F. Aplicação da Metaheurística Busca Tabu ao Problema de Alocação de Aulas a Salas em uma Instituição Universitária. *Revista Produção Online*, v. 11, n. 1, p. 54–75, 2011.
- WALPOLE, R. E.; MYERS, R. H.; MYERS, S. L.; YE, K. *Probability and Statistics for Engineers and Scientists*. New York, NY: Pearson, 2012. 1 – 791 p. ISBN 0132047675.
- WATERER, H. A Zero-one Integer Programming Model for Room Assignment at the University of Auckland. *Proceedings of the 1995 ORSNZ Conference*, Merlborne, Austrália, p. 63–70, 1995.
- WELCH, B. L. The Generalization of Student's Problem when Several Different Population Variances Are Involved. *Biometrika*, v. 34, n. 1-2, p. 28–35, 1947. ISSN 0006-3444.
- ZHOU, A.; QU, B.-Y.; LI, H.; ZHAO, S.-Z.; SUGANTHAN, P. N.; ZHANG, Q. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, Elsevier B.V., v. 1, n. 1, p. 32–49, 2011. ISSN 22106502.

ZITZLER, E.; THIELE, L. Multiobjective optimization using evolutionary algorithms: A comparative case study. *Lecture Notes in Computer Science, Parallel Problem Solving from Nature PPSN V*, v. 1498, p. 292–301, 1998.

ZITZLER, E.; THIELE, L.; LAUMANN, M.; FONSECA, C. M.; FONSECA, V. G. da. Performance assesment of multiobjective optimizers: an analysis and review. *Evolutionary Computation*, v. 7, n. 2, p. 117–132, 2003. ISSN 1089778X.



---



---

# APÊNDICE A

---

## ALGORITMOS DE BUSCA PARA O ABT

As buscas propostas para o ABT são baseadas nos algoritmos mono-objetivo. No Capítulo 5 é apresentada a modificação proposta para a utilização do AGC como busca, no Algoritmo 5.2. Porém para fluidez do texto as modificações propostas para a busca local (Seção 4.2 do Capítulo 4) e BT (Seção 4.3 do Capítulo 4) são discorridas a seguir.

A modificação proposta para o AGC, utiliza toda a população da tabela escolhida  $Tab$ , para iniciar o modelo probabilístico. Para a busca local e BT, porém, deve-se partir de uma solução somente, amostrada de  $Tab$ . A estratégia para gerar um conjunto de soluções será a mesma, i.e., cada vez que a solução corrente for atualizada é também adicionada ao conjunto solução.

A busca local modificada, utiliza a estrutura de vizinhança dada pelo Algoritmo 4.3, no Capítulo 4. Como a busca local só aceita soluções que melhorem a solução corrente, segundo a direção dada, por definição não haverá repetições no conjunto de soluções. O Algoritmo A.1 ilustra a busca proposta.

As entradas do Algoritmo A.1 são a Tabela  $Tab$ , a Direção de Busca  $\vec{D}$  e o tempo de busca  $t_b$ . Na linha 3, verifica-se se a tabela  $Tab$  é a tabela de não-dominadas. Caso positivo, a solução inicial será selecionada aleatoriamente de  $Tab$ , linha 4.

Caso  $Tab$  seja uma tabela de objetivo ou direção de interesse, ela possui uma direção associada  $\vec{D}_{Tab}$ , linha 6. A solução inicial  $X$  será, portanto, a solução da tabela com menor valor nesta direção, linha 7. A comparação, na linha 14, é feita com relação à

---

**Algoritmo A.1:** Busca Local modificada para o ABT
 

---

**Entrada:** Tabela  $Tab$ , Direção de Busca  $\vec{D}$ , Tempo de Busca  $t_b$

**Saída:** Conjunto de Soluções  $\mathbb{Y}$

```

1 Fazer  $t =$  tempo inicial;
2 Fazer  $t_f = t + t_{max}$  tempo final;
3 se  $Tab = Tab_{ND}$  então
4   | Selecionar  $X \in Tab$  aleatoriamente;
5 senão
6   | Seja  $\vec{D}_T$  a direção associada à tabela;
7   | Selecionar  $X = \arg \min_{\bar{X} \in Tab} \{F(\bar{X}, \vec{D}_T)\}$ ;
8 fim
9 Fazer  $\mathbb{Y} = \{X\}$ ;
10 enquanto  $t \leq t_f$  faça
11   | Escolher  $a \in A$  aleatoriamente;
12   | Obter  $X' = \text{MoveAula}(X, a)$ ;
13   | Retornar  $S^a$  para o estado inicial;
14   | se  $F(X', \vec{D}) < F(X, \vec{D})$  então
15     |  $X = X'$ ;
16     |  $\mathbb{Y} = \mathbb{Y} \cup \{X'\}$ ;
17   | fim
18   | Fazer  $t =$  tempo atual;
19 fim
20 retorna  $\mathbb{Y}$ 

```

---

direção de busca  $\vec{D}$ , dada pela expressão (5.4).

Caso a solução corrente seja atualizada, adiciona-a no conjunto de soluções  $\mathbb{Y}$ , na linha 16. A busca continua até que o tempo máximo de busca tenha sido atingido. A saída será o conjunto de soluções  $\mathbb{Y}$ .

A BT modificada irá utilizar as mesmas modificações da busca local, i.e., seleção de elemento baseado no tipo de tabela, introdução do conjunto de soluções  $\mathbb{Y}$  e a incorporação da função  $F(X, \vec{D})$  para comparação. Porém a BT pode repetir a solução corrente, i.e., caso adicione-se todas as atualizações da solução corrente no conjunto  $\mathbb{Y}$ , pode haver repetições. Para que não haja repetições no conjunto  $\mathbb{Y}$ , adiciona-se a solução corrente  $\bar{X}$  em  $\mathbb{Y}$ , somente se  $X \notin \mathbb{Y}$ . O Algoritmo A.2 ilustra o processo.

O parâmetro tamanho máximo da lista tabu foi fixado em  $Tabu = 40$ , como discutido na Seção 6.4 do Capítulo 6. Portanto os parâmetros de entrada para a BT serão a Tabela  $Tab$ , a Direção de Busca  $\vec{D}$  e o tempo de busca  $t_b$ . A diferença notável entre os Algoritmos A.1 e A.2 é a condição, verificada nas linhas 25 e 35, de que a solução corrente não esteja em  $\mathbb{Y}$  para que seja adicionada. O critério de parada é o tempo máximo de busca  $t_b$  e a saída é o conjunto de soluções  $\mathbb{Y}$ .

**Algoritmo A.2:** Busca Tabu modificada para o ABT

---

**Entrada:** Tabela  $Tab$ , Direção de Busca  $\vec{D}$ , Tempo de Busca  $t_b$   
**Saída:** Conjunto de Soluções  $\mathbb{Y}$

- 1 Fazer  $t =$  tempo inicial;
- 2 Fazer  $t_f = t + t_{max}$  tempo final;
- 3 **se**  $Tab = Tab_{ND}$  **então**
- 4 | Selecionar  $X \in Tab$  aleatoriamente;
- 5 **senão**
- 6 | Seja  $\vec{D}_T$  a direção associada à tabela;
- 7 | Selecionar  $X = \arg \min_{\bar{X} \in Tab} \{F(\bar{X}, \vec{D}_T)\}$ ;
- 8 **fim**
- 9 Fazer  $X^* = X$  melhor solução corrente;
- 10 Fazer  $\mathbb{Y} = \{X\}$  ;
- 11 Fazer  $LT = \emptyset$  lista tabu;
- 12 **enquanto**  $t \leq t_f$  **faça**
- 13 | Escolher  $a \in A$  aleatoriamente;
- 14 | Obter  $X' = \text{MoveAula}(X, a)$  ;
- 15 | **se**  $F(X', \vec{D}) < F(X^*, \vec{D})$  **então**
- 16 | |  $X^* = X'$ ;
- 17 | |  $X = X'$ ;
- 18 | |  $LT = \emptyset$ ;
- 19 | |  $\mathbb{Y} = \mathbb{Y} \cup \{X\}$ ;
- 20 | **senão**
- 21 | |  $s = X'(a)$ ;
- 22 | | **se**  $(a, s) \notin LT$  **então**
- 23 | | | **se**  $F(X', \vec{D}) \leq F(X, \vec{D})$  **então**
- 24 | | | |  $X = X'$ ;
- 25 | | | | **se**  $X \notin \mathbb{Y}$  **então**
- 26 | | | | |  $\mathbb{Y} = \mathbb{Y} \cup \{X\}$ ;
- 27 | | | | **fim**
- 28 | | | **senão**
- 29 | | | | **se**  $|LT| = Tabu$  **então**
- 30 | | | | |  $X = X^*$ ;
- 31 | | | | |  $LT = \emptyset$ ;
- 32 | | | | **senão**
- 33 | | | | |  $X = X'$ ;
- 34 | | | | |  $LT = LT + \{(a, s)\}$ ;
- 35 | | | | | **se**  $X \notin \mathbb{Y}$  **então**
- 36 | | | | | |  $\mathbb{Y} = \mathbb{Y} \cup \{X\}$ ;
- 37 | | | | | **fim**
- 38 | | | | **fim**
- 39 | | | **fim**
- 40 | | **fim**
- 41 | **fim**
- 42 | Fazer  $t =$  tempo atual;
- 43 **fim**
- 44 **retorna**  $\mathbb{Y}$

---