

UNIVERSIDADE ESTADUAL PAULISTA “JULIO DE MESQUITA FILHO”

FACULDADE DE ENGENHARIA ELÉTRICA

CÂMPUS DE ILHA SOLTEIRA

Odilon Novaes Silva

**Novas Estratégias de Implementação da Meta-heurística
VNS Aplicada na Otimização de Grade Horária**

Ilha Solteira
2019

Odilon Novaes Silva

**Novas Estratégias de Implementação da Meta-heurística
VNS Aplicada na Otimização de Grade Horária**

Tese de Doutorado apresentada à Faculdade de Engenharia do Câmpus de Ilha Solteira - UNESP como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica. Especialidade: Sistemas de Energia Elétrica

Professor Rubén Augusto Romero Lázaro
Orientador

Ilha Solteira
2019

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

S586n Silva, Odilon Novaes.
Novas estratégias de implementação da meta-heurística VNS aplicada na otimização de grade horária / Odilon Novaes Silva. -- Ilha Solteira: [s.n.], 2019
145 f. : il.

Tese (doutorado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Sistemas de Energia Elétrica, 2019

Orientador: Rubén Augusto Romero Lázaro
Inclui bibliografia

1. Problema de otimização de grade horária. 2. Meta-heurísticas. 3. Meta-heurística VNS. 4. Programação linear binária. 5. Otimização de problemas complexos.


Raiane da Silva Santos

Supervisora Técnica de Seção

Seção Técnica de Referência, Atendimento ao usuário e Documentação
Diretoria Técnica de Biblioteca e Documentação
CRB/8 - 9999

CERTIFICADO DE APROVAÇÃO

TÍTULO DA TESE: Novas estratégias de implementação da meta-heurística VNS na otimização de grade horária

AUTOR: ODILON NOVAES SILVA

ORIENTADOR: RUBEN AUGUSTO ROMERO LAZARO

Aprovado como parte das exigências para obtenção do Título de Doutor em ENGENHARIA ELÉTRICA, área: Automação pela Comissão Examinadora:



Prof. Dr. RUBEN AUGUSTO ROMERO LAZARO
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira



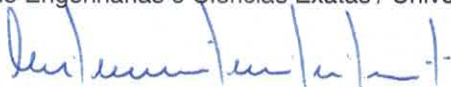
Prof. Dra. ANNA DIVA PLASENCIA LOTUFO
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira



Dr. JONATAS BOAS LEITE
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira



Prof. Dr. CARLOS ROBERTO MENDONÇA DA ROCHA
Centro de Engenharias e Ciências Exatas / Universidade Estadual do Oeste do Paraná - UNIOESTE



Prof. Dr. LUIS GUSTAVO WESZ DA SILVA
Departamento de Indústrias / Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Ilha Solteira, 12 de março de 2019

À minha esposa Cristina, às minhas filhas Lorena, Milena e Patricia. Às minhas netas Eloah, Mariah e Agatha e aos meus pais Odilon e Alvina que me educaram e me possibilitaram mais essa conquista, exemplos de vida fundamentais para minha vida pessoal e profissional.

Agradecimentos

- A Deus por ter me dado a saúde e a fé necessários para a conclusão de mais uma jornada.
- Aos meus pais, Odilon e Alvina, que durante a passagem abreviada dessa vida sempre me apoiaram em todos os momentos. Agradeço por todo o amor e carinho.
- À minha esposa Cristina e minhas filhas Lorena, Milena e Patricia, pelo incentivo, carinho, paciência e compreensão de sempre, fundamentais para o meu envolvimento nesse trabalho.
- Aos meus sobrinhos Beto, Gilmar Jr. e Suze pelo espírito de amizade e companheirismo.
- Ao compadre Gilmar pelo incentivo e envolvimento na concretização deste sonho, meus sinceros agradecimentos.
- A todos os meus irmãos, irmãs, cunhadas, cunhados, sobrinhas, sobrinhos e genros meus sinceros agradecimentos pelo carinho e pelas palavras de incentivo.
- Agradeço em especial ao Professor Rubén que abriu as portas e acolheu-me. A este orientador e amigo expresso minha profunda gratidão pelo incentivo, envolvimento, conselhos e críticas, os quais contribuíram para o meu crescimento pessoal e profissional, obrigado por acreditar em mim.
- Ao Instituto Federal do Mato Grosso do Sul.
- Aos professores e funcionários do DEE e aos colegas do LAPSEE, especialmente ao colega Leonardo Macedo Possagnolo.
- Meus sinceros agradecimentos aos amigos, em especial aos que me acompanharam de perto nas dificuldades e realizações nesse tempo. Meus companheiros do grupo “Cuiabá forte em Ilha”.
- A todos aqueles que com um gesto ou uma palavra contribuíram para a realização deste trabalho e cujos nomes não aparecem, agradeço de igual modo e perdoem-me pelo meu esquecimento.

Finalmente, agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e à FAPESP pelo apoio à pesquisa através do Projeto Temático No. 2015/21972-6.

*“Que os vossos esforços desafiem as impossibilidades,
lembrai-vos de que as grandes coisas do homem foram
conquistadas do que parecia impossível”.*

Charles Chaplin.

Resumo

Neste projeto de pesquisa, é abordado o problema de otimização de grade horária. O tipo de problema de grade horária abordado é aquele que tem o enunciado e a estrutura de dados apresentado no site da Competição Internacional de Otimização do Problema de Grade Horária. Esse problema pode ser modelado como sendo um problema de Programação Linear Binária de grande porte. Entretanto, os solvers comerciais disponíveis, como o CPLEX, não tem a capacidade de encontrar as soluções ótimas das 20 instâncias mostradas no site da Competição Internacional de Otimização do Problema de Grade Horária. Neste trabalho foi desenvolvido um algoritmo VNS especializado para resolver o problema de otimização de grade horária. A parcela inovadora da proposta está relacionado com o uso da lógica de partição para encontrar a melhor solução vizinha da solução corrente de forma eficiente e para uma estrutura de vizinhança complexa e formada por muitos elementos. Dessa forma, a proposta de otimização se tornou muito eficiente na resolução das 20 instâncias cujos dados se encontram no site da Competição Internacional de Otimização do Problema de Grade Horária.

Palavras Chave: Problema de otimização de grade horária. Meta-heurísticas. Meta-heurística VNS. Programação linear binária. Otimização de problemas complexos.

Abstract

In this research project, we address the optimization timetabling problem. The type of timetabling problem addressed is one that has the statement and data structure displayed on the site of the International Competition of Optimization of the Timetabling Problem. This problem can be modeled as a large Binary Linear Programming Problem. However, the commercial solvers available, such as CPLEX, do not have the ability to find the optimal solutions from the 20 instances shown on the site of the International Competition of Optimization of the timetabling Problem. In this work a specialized VNS algorithm was developed to solve the optimization of Timetabling Problem . The innovative part of the proposal is related to the use of partition logic to find the best neighborhood solution of the current solution efficiently and to a structure of complex neighborhood formed by many elements. In this way, the optimization proposal became very efficient in the resolution of the 20 instances whose data were found on the website of the International Competition for Optimization of the Timetabling Problem.

Keywords: Timetabling optimization problem. Meta-heuristics. Meta-heuristic VNS. Linear binary programming. Optimization of complex problems.

LISTA DE FIGURAS

Figura 1: Representação de uma proposta de solução	59
Figura 2: Vetor a e matriz A do banco de dados	62
Figura 3: Vetor b e matriz B do banco de dados	62
Figura 4: Vetor c e matriz C do banco de dados	64
Figura 5: Programação dos eventos do aluno No. 1	65

LISTA DE TABELAS

Tabela 1: Banco de dados típico do problema de grade horária	17
Tabela 2: Proposta de solução típica do problema de grade horária	19
Tabela 3: Segunda proposta de representar uma proposta de solução	61
Tabela 4: Instâncias dos problemas de grade horária	73
Tabela 5: Melhores resultados encontrados	74
Tabela 6: Melhor solução encontrada para a Instância No. 1	76
Tabela 7: Melhor solução encontrada para a Instância No. 2	77
Tabela 8: Melhor solução encontrada para a Instância No. 3	78
Tabela 9: Melhor solução encontrada para a Instância No. 4	79
Tabela 10: Melhor solução encontrada para a Instância No. 5	80
Tabela 11: Melhor solução encontrada para a Instância No. 6	81
Tabela 12: Melhor solução encontrada para a Instância No. 7	82
Tabela 13: Melhor solução encontrada para a Instância No. 8	83
Tabela 14: Melhor solução encontrada para a Instância No. 9	84
Tabela 15: Melhor solução encontrada para a Instância No. 10	85
Tabela 16: Melhor solução encontrada para a Instância No. 11	86
Tabela 17: Melhor solução encontrada para a Instância No. 12	87
Tabela 18: Melhor solução encontrada para a Instância No. 13	88
Tabela 19: Melhor solução encontrada para a Instância No. 14	89
Tabela 20: Melhor solução encontrada para a Instância No. 15	90
Tabela 21: Melhor solução encontrada para a Instância No. 16	91
Tabela 22: Melhor solução encontrada para a Instância No. 17	92
Tabela 23: Melhor solução encontrada para a Instância No. 18	93
Tabela 24: Melhor solução encontrada para a Instância No. 19	94
Tabela 25: Melhor solução encontrada para a Instância No. 20	95
Tabela 26: Banco de dados da Instância No. 1	105
Tabela 27: Banco de dados da Instância No. 2	114
Tabela 28: Banco de dados da Instância No. 3	123
Tabela 29: Banco de dados da Instância No. 4	132

SUMÁRIO

1. Introdução Geral	12
2. O Problema de Otimização de Grade Horária	16
2.1 Dados do Problema de Grade Horária	16
2.2 Função Objetivo e Restrições no Problema de Grade Horária	17
2.3 Estrutura da Representação de uma Proposta de Solução	18
2.4 Modelagem Matemática do Problema de Otimização de Grade Horária	20
2.5 Revisão Bibliográfica do Problema de Otimização de Grade Horária	23
3. Introdução sobre as Meta-heurísticas	28
3.1 Aspectos Fundamentais Relacionados com as Heurísticas e as Meta-heurísticas	28
3.2 Revisão sobre as Heurísticas	29
3.2.1 O Algoritmo Heurístico Construtivo	30
3.2.2 O Algoritmo Heurístico de Busca Através de Vizinhança	31
3.3 Revisão sobre as Meta-heurísticas	35
3.3.1 <i>Simulated Annealing</i>	35
3.3.2 <i>Tabu Search</i> - Busca Tabu	37
3.3.3 O Algoritmo Genético	40
3.3.4 A Meta-heurística GRASP	43

4. A Meta-heurística VNS	48
4.1 O Algoritmo VND	50
4.2 O Algoritmo RVNS	52
4.3 O Algoritmo BVNS	54
4.4 O Algoritmo GVNS	55
5. Meta-heurística VNS Aplicado ao Problema de otimização de Grade Horária	58
5.1 Introdução	58
5.2 Forma de Representação de uma Proposta de Solução	59
5.3 Verificação das Restrições Obrigatórias e das Restrições Secundárias	63
5.4 Estrutura VNS Proposta	66
6. Resultados de Testes	72
6.1 Introdução	72
7. Conclusões	96
A. Dados das Instâncias Usadas em Testes	104

Capítulo 1

Introdução Geral

Nesta tese de doutorado, é abordado o problema de otimização de grade horária no ambiente universitário. Esse tipo de problema corresponde a um ramo de pesquisa muito ativo na Pesquisa Operacional e considera todos os problemas em que é necessária a programação de atividades em um contexto em que existem ambientes disponíveis para a programação de atividades e com características muito bem definidas, existem horários bem definidos para a ocupação desses ambientes e existem turmas que precisam usar esses ambientes. Esse tipo de problema é conhecido como *timetabling* na literatura especializada de Pesquisa Operacional.

Os problemas da família *timetabling* são muito variados e podem ser usados em contextos também muito variados, isto é, em ambientes de ensino como nas escolas pré-universitárias, nas universidades e em ambientes de capacitação em geral. Em geral, nos problemas do tipo *timetabling* pode-se considerar a otimização da grade horária levando em conta restrições de tipo variado tais como disponibilidade, capacidade e características especiais da sala de aula, disponibilidade de alunos ou de turmas de estudantes, disponibilidade de professores dentro de uma grade horária, tipicamente semanal, entre outros. O resultado final do processo de otimização é uma grade horária que otimiza uma função objetivo especificada.

O problema de programação ótima de horários geralmente é muito difícil de modelar matematicamente, isto é, nem sempre é possível encontrar um modelo matemático de otimização de forma que seja possível resolver o problema usando um *solver* de otimização. Em alguns casos é possível desenvolver um modelo matemático de otimização, mas o tempo necessário para resolver instâncias típicas pode ser muito elevado e em muitos casos o *solver* de otimização geralmente termina o processo de otimização antes de encontrar a solução ótima por problemas de memória para armazenamento das informações e fornecendo como proposta de solução uma solução apenas de boa qualidade, sem a garantia de que essa solução seja ótima, com a informação do *gap* de otimização, isto é, uma medida da diferença que existe entre a solução fornecida e a limitante inferior.

Neste trabalho, o tipo de problema de otimização de grade horária escolhido é a proposta formulada pela competição internacional de grade horária. O motivo da escolha é que, trata-se de um tipo de problema muito difícil de resolver e, portanto, muito desafiador. Também no

site correspondente existem informações muito valiosas. Assim, existe disponível a informação completa de 20 instâncias (dados completos de 20 casos específicos para resolver) e a informação das melhores propostas de solução encontradas por diferentes técnicas de otimização. Adicionalmente, no site existe a possibilidade de verificar se uma certa proposta de solução encontrada por uma técnica de otimização está correta, isto é, se a proposta de solução encontrada é factível e o valor correspondente da função objetivo.

No problema de otimização de grade horária abordado neste trabalho os dados estão representados da seguinte forma:

- Existem eventos especificados. Um evento pode ser uma aula de uma hora de duração de uma certa disciplina. Em cada evento existem alunos matriculados e, portanto, existe a informação dos alunos matriculados em cada evento.
- Existem salas de aula com características especificadas. Essas características definem os eventos que podem ser programados em cada tipo de sala.
- Existem alunos com a informação dos eventos nos quais se encontra matriculado cada aluno.
- Existe uma grade horária disponível para programar os eventos, isto é, os dias disponíveis por semana e o número de horas disponíveis por dia em que podem ser programados os eventos.

Adicionalmente, o problema de grade horária abordado apresenta dois tipos de restrições, isto é, restrições obrigatórias que devem ser necessariamente satisfeitas para que uma proposta de solução seja considerada factível e restrições secundárias. Uma proposta de solução é considerada factível se não existem restrições obrigatórias violadas e essa proposta é considerada ótima se não apresenta restrições secundárias violadas. Adicionalmente, uma proposta de solução é melhor que outra se apresenta um menor número de restrições secundárias violadas.

As restrições obrigatórias são as seguintes:

1. Dois ou mais eventos não podem ser programados na mesma sala e no mesmo horário.
2. A sala programada para um evento deve ter as características requeridas pelo evento e a capacidade suficiente para acomodar os estudantes matriculados no evento (disciplina).
3. Um estudante não pode ser programado em mais de um evento no mesmo horário.

As restrições secundárias são as seguintes:

1. Um estudante não deveria ser programado em mais de dois eventos consecutivos, isto é, não é desejável que um estudante seja programado em 3 ou mais eventos consecutivos.

2. Um estudante não deveria ser programado em apenas um evento em um dia.
3. Um estudante não deveria ser programado no último horário do dia.

Considerando que o desafio consiste em resolver as 20 instâncias existentes no banco de dados da Competição Internacional de Grade Horária, então o problema de grade horária é um problema de difícil resolução. A dificuldade fica evidente se for levado em conta os seguintes fatos:

- Cada instância está constituído por 400 eventos que devem ser alocados em uma grade horária que possui 450 células. Em outras palavras, deve-se alocar de forma não conflitante 400 elementos em uma matriz de 450 elementos (88,9% de taxa de ocupação). Se considerarmos que não é recomendável programar eventos na última hora do dia, então o problema consiste em alocar 400 elementos em uma matriz de 400 elementos e sem nenhum tipo de conflito (100% de taxa de ocupação).
- Todas as instâncias foram propositalmente montadas de forma que a função objetivo $v = 0$, isto é, de forma que 400 elementos sejam alocados em uma matriz de 400 elementos e sem nenhum tipo de conflito. Esse fato significa que se existe um modelo matemático de programação linear inteira mista para esse tipo de problema de otimização de grade horária e se esse modelo for resolvido usando um solver de otimização como o CPLEX, então em qualquer instante anterior da identificação da solução ótima o *gap* de convergência deve ser de 100%. Esse fato, adicionalmente significa que os testes de sondagem de um algoritmo tipo *branch and bound*, que representa a base principal de um solver do tipo CPLEX, em determinadas condições se tornam ineficientes. Finalmente, se o solver como o CPLEX termina o processamento de forma prematura por problemas de memória para armazenamento, então o CPLEX termina o processo informando a melhor solução encontrada e com *gap* igual a 100%.
- Pesquisadores do Laboratório de Planejamento de Sistemas de Energia Elétrica (LAPSEE) desenvolveram um modelo matemático de programação linear inteira mista para o problema de grade horária abordado neste trabalho. Assim, foi usado esse modelo e o solver CPLEX para resolver as 20 instâncias do Sistema Internacional de Competição de Grade Horária. Em todos os casos o CPLEX terminou o processo de solução sem convergência por problemas de memória para armazenamento da informação. Entretanto, esse modelo foi usado como elemento auxiliar na meta-heurística VNS implementado neste trabalho como é mostrado adiante.

Em problemas em que não existem modelos matemáticos ou esses modelos matemáticos exigem um tempo de processamento inaceitável usando os melhores *solvers* existentes, então esse tipo de problemas pode ser resolvido usando heurísticas e meta-heurísticas. Esse é o caso do problema de grade horária abordado neste trabalho. Assim, neste trabalho é desenvolvido um algoritmo VNS altamente especializado para resolver o problema de grade horária. A meta-heurística VNS já se mostrou eficiente na resolução de muitos problemas existentes na literatura de pesquisa operacional e por esse motivo foi escolhida neste trabalho.

Este trabalho está dividido em 5 capítulos. No Capítulo 1 é realizada uma introdução geral do problema de grade horária abordado neste trabalho. No Capítulo 2 é apresentado em detalhes o problema de grade horária, é apresentado um modelo matemático desenvolvido no LAPSEE e também é feita uma revisão bibliográfica geral sobre os problemas de grade horária. também nesse capítulo é discutido sobre as características das instâncias resolvidas. No Capítulo 3 são abordados os tópicos relacionados com as heurísticas e as meta-heurísticas aplicados a problemas complexos. A abordagem prioriza a lógica de funcionamento das heurísticas e as meta-heurísticas e as características específicas das principais heurísticas e meta-heurísticas. No Capítulo 4 é abordado em detalhes a meta-heurística VNS usada neste trabalho. Como essa meta-heurística foi escolhida como técnica de otimização neste trabalho, então a abordagem é realizada de forma detalhada, analisando as qualidades e os defeitos da meta-heurística. No Capítulo 5 é formulada a meta-heurística VNS especializada para resolver o problema de grade horária. A forma de idealizar uma proposta de solução para o problema de grade horária, as formas de idealizar as estruturas de vizinhança e a forma de gerar soluções factíveis (sem violar as restrições obrigatórias) são de grande importância na idealização do algoritmo VNS especializado. No Capítulo 6 são realizados os testes para as 20 instâncias e uma análise comparativa com outras propostas de otimização para o mesmo problema. No Capítulo 7 são resumidas as principais conclusões e no Apêndice são mostrados os dados das 4 instâncias usadas nos testes.

Capítulo 2

O Problema de Otimização de Grade Horária

Neste capítulo são abordados os principais tópicos relacionados com o tipo de problema de otimização de grade horária analisado no trabalho. Assim, inicialmente são mostrados os dados necessários e a estrutura em que esses dados são apresentados da forma mais adequada. A seguir é apresentada a formulação do problema, mencionando a forma da função objetivo e as restrições obrigatórias e secundárias. Na sequência é apresentada a proposta mais adequada de armazenar e apresentar uma proposta de solução. Depois é apresentada a modelagem matemática de programação linear inteira mista, desenvolvida no LAPSEE, para o problema de otimização de grade horária. Finalmente é realizada uma revisão bibliográfica das principais publicações relacionadas com o problema de programação de grade horária de tipo geral.

2.1 Dados do Problema de Grade Horária

Como mencionado de forma preliminar anteriormente, o banco de dados está formado por um número de eventos (cada evento pode ser uma disciplina ou uma componente de uma disciplina), para cada evento existem alunos matriculados e, portanto, existem um número conhecido de alunos especificado. Adicionalmente, existem salas de aula em número conhecido e cada evento pode ser programado apenas em algumas salas que cumprem os requisitos exigidos pelo evento. Assim, no problema de otimização de grade horária abordado neste trabalho os dados estão representados da seguinte forma:

1. Existem eventos especificados. Um evento pode ser uma aula de uma hora de duração de uma certa disciplina. Em cada evento existem alunos matriculados e, portanto, existe a informação dos alunos matriculados em cada evento. Nas instâncias usadas neste trabalho existem 400 eventos a serem programados.
2. Existem salas de aula com características especificadas. Essas características definem os eventos que podem ser programados em cada tipo de sala. Nas instâncias usadas neste trabalho existem 10 salas de aula.

3. Existem alunos com a informação dos eventos nos quais se encontra matriculado cada aluno. Nas instâncias usadas neste trabalho existem 200 alunos.
4. Existe uma grade horária disponível para programar os eventos, isto é, os dias disponíveis por semana e o número de horas disponíveis por dia em que podem ser programados os eventos. Nas instâncias usadas neste trabalho existem 5 dias para programar as aulas e cada dia existem 9 horas consecutivas para programar os eventos.

Na Tabela 1 é mostrado um caso típico de banco de dados que representa a informação parcial da Instância No. 2 do banco de dados da Competição Internacional de Grade Horária. Nesse caso existem 400 eventos que devem ser programados, 10 salas de aula disponíveis e 200 alunos. Assim, por exemplo, o evento 5 pode ser programada em 8 salas de aula das 10 salas de aula disponíveis (as salas 1, 4, 5, 6, 7, 8, 9 e 10) e nesse evento se encontram matriculados 11 alunos (os alunos 29, 30, 31, 32, 33, 34, 35, 36, 37, 39 e 101). Adicionalmente, existe a informação de que podem ser programados eventos em cinco dias da semana e 9 horas de atividades consecutivas por dia. Portanto, existem 45 alternativas de programação horária para cada evento.

Tabela 1: Banco de dados típico do problema de grade horária.

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos alunos
1	3	3, 7, 8	10	3, 142, 144, 146, 148, 150, 152, 160, 164, 184
2	1	8	12	34, 35, 36, 39, 41, 43, 45, 47, 61, 64, 99, 171
3	1	10	12	91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 102, 172
4	2	2, 5	2	75, 76
5	8	1, 4, 5, 6, 7, 8, 9, 10	11	29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 101
6	2	4, 7	11	29, 30, 31, 32, 33, 34, 35, 36, 37, 40, 108
7	2	5, 6	10	2, 10, 14, 18, 22, 109, 194, 195, 198, 200
8	2	5, 10	7	28, 29, 30, 31, 32, 41, 110
9	1	1	11	144, 146, 148, 150, 152, 154, 156, 158, 162, 164, 172
...
400	1	7	11	4, 8, 99, 171, 192, 193, 194, 195, 196, 197, 200

Fonte: Elaborado pelo autor.

2.2 Função Objetivo e Restrições no Problema de Grade Horária

O objetivo do problema de grade horária é encontrar uma proposta de solução que cumpra com todas as restrições obrigatórias e minimize as restrições secundárias. Assim, a solução ótima é aquela proposta de solução que tem o menor número de restrições secundárias violadas. Como mencionado anteriormente, nas 20 instâncias analisadas sobre o problema da elaboração ótima de grade horária a solução ótima cumpre também com todas as restrições secundárias. Em outras palavras, as instâncias foram propositalmente construídas para cumprir com todas

as restrições secundárias. Entretanto, na formulação do modelo matemático de otimização o objetivo é minimizar o número de restrições secundárias sujeito a cumprir com todas as restrições obrigatórias.

Neste trabalho, de acordo com a formulação da Competição Internacional de Grade horária, as restrições obrigatórias são as seguintes:

1. Dois ou mais eventos não podem ser programados na mesma sala e no mesmo horário. Este tipo de restrição chamaremos de RO_1 neste trabalho.
2. A sala programada para um evento deve ter as características requeridas pelo evento e a capacidade suficiente para acomodar os estudantes matriculados no evento (disciplina). Este tipo de restrição chamaremos de RO_2 neste trabalho.
3. Um estudante não pode ser programado em mais de um evento no mesmo horário. Este tipo de restrição chamaremos de RO_3 neste trabalho.

Neste trabalho, também de acordo com a formulação da Competição Internacional de Grade horária, as restrições secundárias são as seguintes:

1. Um estudante não deveria ser programado em mais de dois eventos consecutivos, isto é, não é desejável que um estudante seja programado em 3 ou mais eventos consecutivos. Este tipo de restrição chamaremos de RD_1 neste projeto.
2. Um estudante não deveria ser programado em apenas um evento em um dia. Este tipo de restrição chamaremos de RD_2 neste projeto.
3. Um estudante não deveria ser programado no último horário do dia. Este tipo de restrição chamaremos de RD_3 neste projeto.

2.3 Estrutura da Representação de uma Proposta de Solução

A solução do problema de grade horária pode ser representada em um arranjo matricial do tamanho $nh \times ns$, sendo nh o número de horas e ns o número de salas. Um elemento desse arranjo matricial indica um evento. Assim, seja R a matriz que mostra a grade horária, então o elemento $R(i, j)$ indica o evento que foi programado no horário i e na sala j . Nas instâncias analisadas e testadas para o problema de grade horária neste trabalho existem 45 horas de programação (5 dias de programação e 9 horas programadas por dia) e 10 salas de aula.

Na Tabela 2 é mostrada uma proposta de solução para a instância No. 2 do banco de dados da Competição Internacional de Grade Horária. Nessa tabela, o primeiro dia (segunda feira) está representada pela numeração horária de 1-9, a numeração 10-18 corresponde ao segundo dia e assim em diante. Nesse contexto, pode-se verificar que na proposta de solução apresentada na

Tabela 2: Proposta de solução típica do problema de grade horária.Solução factível: $v = 318$ (RD1 = 150, RD2 = 15, RD3 = 153)

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	57	146	371	133	229	111	400	198	142	27
2	105	265	203	31	179	187	251	354	88	244
3	166	228	75	206	316	360	0	63	107	144
4	197	0	150	167	384	273	6	196	300	231
5	157	0	103	131	52	99	10	114	292	49
6	243	165	59	293	22	226	141	124	398	37
7	36	0	186	129	359	217	204	308	84	125
8	201	285	136	175	312	259	223	173	329	134
9	73	0	151	0	234	0	85	0	0	0
10	45	20	236	139	388	149	350	325	115	336
11	120	272	326	380	182	289	177	335	208	0
12	81	42	366	72	156	392	30	288	355	46
13	51	352	282	337	40	353	44	256	172	0
14	48	19	91	215	60	393	50	330	68	277
15	98	333	95	0	351	374	188	320	109	112
16	321	318	155	205	90	0	87	268	168	239
17	304	121	147	94	389	395	93	270	263	5
18	250	0	241	0	385	0	212	0	178	0
19	346	307	61	246	369	0	342	299	301	224
20	32	43	378	331	339	34	56	284	78	233
21	306	219	154	379	18	117	199	140	169	213
22	108	261	101	262	41	74	104	116	319	0
23	193	174	279	96	254	7	0	313	322	358
24	189	309	181	338	387	347	70	323	100	14
25	275	274	344	377	164	305	123	16	345	200
26	327	207	396	110	185	145	148	221	399	17
27	0	0	0	0	69	356	0	240	55	291
28	128	191	218	317	135	390	162	302	283	97
29	161	132	368	266	383	11	1	2	80	290
30	210	92	66	122	24	252	0	286	0	3
31	235	138	211	314	15	397	26	334	247	357
32	9	365	255	375	281	348	361	71	220	0
33	362	47	260	113	382	183	195	267	13	0
34	311	4	372	106	367	0	248	297	143	153
35	184	119	310	238	202	269	33	82	216	194
36	0	0	25	242	237	192	0	127	0	0
37	278	287	86	376	163	394	214	315	137	190
38	363	328	332	257	245	158	227	130	67	170
39	209	222	370	180	271	391	159	303	0	295
40	65	343	0	58	340	12	21	79	23	29
41	296	324	373	39	386	230	253	0	38	225
42	76	280	160	64	126	118	54	0	53	102
43	62	298	258	364	176	0	35	77	232	341
44	264	276	381	249	28	349	0	171	89	152
45	0	0	0	0	0	0	0	0	294	83

Fonte: Elaborado pelo autor.

Tabela No. 2 existem poucos eventos programados nas horas 9, 18, 27, 36 e 45 que correspondem à última hora de aula de cada dia programado (segunda a sexta feira). Essa característica da proposta de solução está relacionada com o fato de que programar eventos na última hora de um dia viola a restrição secundária RD3.

A estrutura matricial usada para representar uma proposta de solução, como mostrada na Tabela No. 2, também vai ser adotada para representar uma proposta de solução na implementação das meta-heurísticas para o problema de otimização de grade horária.

2.4 Modelagem Matemática do Problema de Otimização de Grade Horária

No LAPSEE foi desenvolvido um modelo matemático de otimização para o problema de grade horária que pode resolver problemas com as características da Competição Internacional de Grade Horária. Neste trabalho, esse modelo é usado para tentar resolver as instâncias existentes no página web da Competição Internacional de Grade Horária com a ideia de avaliar o desempenho desse modelo matemático quando resolvido por *solvers* comerciais. Entretanto, esse modelo também é usado dentro da estrutura da meta-heurística VNS desenvolvido neste trabalho. A seguir é apresentado o modelo matemático.

O modelo matemático assume a seguinte forma:

$$\text{Min } v = \sum_{a \in \Omega_A} \sum_{d=1}^5 (f_{a,d} + u_{a,d} + \sum_{k=1}^7 c_{a,d,k}) \quad (1)$$

s.a.

$$\sum_{s \in \Omega_s} \sum_{t \in \Omega_t} x_{e,s,t} = 1 \quad \forall e \in \Omega_E \quad (2)$$

$$\sum_{e \in \Omega_E} x_{e,s,t} \leq 1 \quad \forall s \in \Omega_s; t = 1, 2, \dots, 45 \quad (3)$$

$$ES_{e,s} - x_{e,s,t} \geq 0 \quad \forall e \in \Omega_E, s \in \Omega_s; t = 1, 2, \dots, 45 \quad (4)$$

$$HA_{a,t} = \sum_{e \in \Omega_E} EA_{e,a} \sum_{s \in \Omega_s} x_{e,s,t} \quad \forall a \in \Omega_A; t = 1, 2, \dots, 45 \quad (5)$$

$$HA_{a,t} \leq 1 \quad \forall a \in \Omega_A; t = 1, 2, \dots, 45 \quad (6)$$

$$\sum_{t=9d+k-9}^{9d+k-7} HA_{a,t} \leq 2 + c_{a,d,k} \quad \forall a \in \Omega_A; d = 1, 2, \dots, 5 \quad k = 1, 2, \dots, 7 \quad (7)$$

$$f_{a,d} = HA_{a,9d} \quad \forall a \in \Omega_A; d = 1, 2, \dots, 5 \quad (8)$$

$$\sum_{t=9d-8}^{9d} HA_{a,t} = u_{a,d} + n_{a,d} \quad \forall a \in \Omega_A; d = 1, 2, \dots, 5 \quad (9)$$

$$\sum_{t=9d-8}^{9d} HA_{a,t} \geq 2\alpha_{a,d} \quad \forall a \in \Omega_A; d = 1, 2, \dots, 5 \quad (10)$$

$$2\alpha_{a,d} \leq n_{a,d} \leq 9\alpha_{a,d} \quad \forall a \in \Omega_A; d = 1, 2, \dots, 5 \quad (11)$$

$$u_{a,d} \geq 0 \quad \forall a \in \Omega_A; d = 1, 2, \dots, 5 \quad (12)$$

$$\alpha_{a,d} \in \{0, 1\} \quad \forall a \in \Omega_A; d = 1, 2, \dots, 5 \quad (13)$$

$$x_{e,s,t} \in \{0, 1\} \quad \forall e \in \Omega_E; s \in \Omega_s; t = 1, 2, \dots, 45 \quad (14)$$

No modelo matemático apresentado anteriormente existem as seguintes variáveis de decisão:

- $x_{e,s,t}$ é uma variável de decisão binária que indica que $x_{e,s,t} = 1$ se o evento e está programado na sala s e no bloco de tempo t e vale $x_{e,s,t} = 0$ em caso contrário. Assim, para um problema típico usado neste trabalho existem 400 eventos, 10 salas de aula e 45 blocos de tempo e, portanto, nesse caso existem 180.000 variáveis binárias desse tipo.
- $\alpha_{a,d}$ é uma variável de decisão binária que indica que $\alpha_{a,d} = 1$ se o aluno a está programado no dia d e vale $\alpha_{a,d} = 0$ em caso contrário. Assim, para um problema típico usado neste trabalho existem 200 alunos e 5 dias da semana e, portanto, nesse caso existem 1.000 variáveis binárias desse tipo.
- A variável $HA_{a,t}$ determina o estado de ocupação do aluno a no bloco de tempo t . Assim, o valor dessa variável depende do valor de $x_{e,s,t}$ no bloco de tempo t .

No modelo matemático apresentado anteriormente existem os seguintes conjuntos:

- Ω_E é o conjunto de eventos. Nos problemas analisados geralmente existem 400 eventos e, portanto, $|\Omega_E| = 400$.
- Ω_s é o conjunto de salas de aula. Nos problemas analisados geralmente existem 10 salas de aula e, portanto, $|\Omega_s| = 10$.
- Ω_t é o conjunto de blocos de tempo. Nos problemas analisados geralmente existem 45 blocos de tempo e, portanto, $|\Omega_t| = 45$.
- Ω_A é o conjunto de alunos. Nos problemas analisados geralmente existem 200 alunos e, portanto, $|\Omega_A| = 200$.

No modelo matemático apresentado anteriormente existem os seguintes parâmetros:

- A matriz ES armazena a habilitação de uma sala para acolher um evento. ES é uma matriz de elementos binários obtido do banco de dados. Assim, $ES_{e,s} = 1$ indica que o evento e pode ser programado na sala s e $ES_{e,s} = 0$ indica que o evento e não pode ser programado na sala s .

- A matriz EA armazena a informação de que um aluno se encontra matriculado em um evento e a informação também se encontra armazenada em forma binária a partir do banco de dados. Assim, $EA_{e,a} = 1$ indica que o aluno a está matriculado no evento e e $EA_{e,a} = 0$ indica que o aluno a não está matriculado no evento e .

No modelo matemático apresentado anteriormente existem as seguintes restrições:

- A restrição (2) garante que cada evento deve ser programado, necessariamente, em apenas uma sala de aula e em um bloco de tempo. Como normalmente existem 400 eventos, então existem 400 restrições desse tipo. Essa restrição garante que todos os eventos devem ser programados.
- A restrição (3) permite que para cada sala de aula e para um bloco de tempo pode ser programado apenas um evento ou nenhum evento. Como normalmente existem 10 salas de aula e 45 blocos de tempo, então existem 450 restrições desse tipo. Essa restrição garante que na mesma sala e no mesmo bloco de tempo não pode ser programado mais de um evento (restrição obrigatória RO_1).
- A restrição (4) permite que um evento seja programado em uma sala de aula adequada e habilitada. Deve-se observar que para que um evento seja programado em uma sala, então $ES(e, s) = 1$ e portanto, apenas nesse contexto o evento pode ser programado nessa sala e $x(e, s, t)$ pode assumir um valor de $x(e, s, t) = 1$, isto é, o evento pode ser programado nessa sala em algum bloco de tempo. Essa restrição garante que um evento deve ser programado em uma sala de aula adequada (restrição obrigatória RO_2).
- A restrição (5) calcula o valor de $HA_{a,t}$, isto é, o estado do horário do aluno a no bloco de tempo t . A restrição (6) determina que esse valor deve ser menor que 1, isto é, no horário do aluno a no bloco de tempo t deve aparecer como programado ($HA_{a,t} = 1$) ou não programado ($HA_{a,t} = 0$). Essas restrições garantem o cumprimento da restrição obrigatória RO_3 .
- A restrição (7) permite verificar se um aluno foi programado em mais de dois eventos consecutivos. Quando um aluno for programado em mais de dois eventos consecutivos, então a variável $c_{a,d,k}$ assume o valor de $c_{a,d,k} = 1$. Assim, essa restrição avalia três horas consecutivas e verifica se um aluno está programado em mais de dois eventos. Dessa forma em cada dia existem sete combinações que devem ser avaliadas. Portanto, essa restrição permite quantificar as restrições secundárias RD_1 .
- A restrição (8) permite verificar se um aluno foi programado no último horário de um dia através da variável $f_{a,d}$. Portanto, essa restrição permite quantificar as restrições secundárias RD_3 .
- As restrições (9)-(13) permitem verificar se um aluno foi programado em apenas uma aula em um dia. Nessas restrições, a variável $n_{a,d}$ se encontra ativa e variando entre 2 e 9 quando a soma de eventos no dia d para o aluno a é maior que 2. Esse valor é controlado pela variável binária $\alpha_{a,d}$. Portanto, esse conjunto de restrições permite determinar o valor das restrições secundárias RD_2 violadas através da variável $u_{a,d}$.

O modelo matemático do problema de otimização de grade horária apresentado é um problema de programação linear inteira mista (PLIM). Portanto, pode ser resolvido usando um *solver* comercial como o CPLEX, disponível no LAPSEE. Entretanto, para as 20 instâncias da Competição Internacional de Grade Horária, o CPLEX termina o processo antes da convergência por problemas de armazenamento de memória para informação. Também, como foi mencionado anteriormente, o modelo apresentado é usado de forma restrita na estrutura VNS desenvolvido neste trabalho.

2.5 Revisão Bibliográfica do Problema de Otimização de Grade Horária

A literatura existente sobre o problema de programação ótima de grade horária é grande e diversificada. Entretanto, existem muitos tipos de problemas de grade horária, muitos deles resolvem problemas específicos e, portanto, o tipo de formulação e o tipo de dados são diferentes dos existentes no banco de dados da Competição Internacional de Otimização de Grade Horária. Assim, apresenta-se uma revisão de alguns desses trabalhos.

O trabalho de FERLAND et al. (1992), apresenta uma modelagem de programação linear binária para problemas de programação de horários, de modo que o tratamento de problemas como o Problema de Programação de horários Professor \times Turma, chamado de PPT, é possível através da utilização das chamadas restrições laterais (*side constraints*). Essa modelagem é relativamente genérica e permite a modelagem de várias restrições fortes, mas não permite a modelagem de problemas com restrições fracas, como as existentes no Problema de Programação de Horários Professor \times Turma de forma compacta, chamado de PPTC. Um procedimento heurístico baseado em busca local é apresentado. Nesse procedimento, a busca local primeiramente é aplicada para a diminuição progressiva das infactibilidades. Logo que uma solução factível é obtida, inicia-se a busca por melhores soluções. Três métodos são apresentados: no primeiro utiliza trocas simples como busca local; o segundo procedimento realiza uma busca recursiva que apresenta similaridades em relação ao método de busca tabu e, finalmente, um procedimento de relaxação lagrangeana é apresentado.

O método de Busca Tabu, proposto por GLOVER (1986) e também uma proposta semelhante proposta por HANSEN (1986), faz uso explícito de estruturas de memória para generalizar a heurística de busca através de vizinhança, de modo que esses continuem a exploração mesmo na ausência de uma solução vizinha de melhor qualidade. O trabalho de HERTZ (1991), desenvolve métodos de busca tabu para a geração de horários em universidades, baseados em uma implementação prévia do método para coloração de grafos. Uma peculiaridade dessa proposta é a decisão de colocar a violação de algumas restrições fracas fora do espaço de busca (qualquer solução sempre cumpre com essas exigências) e tentar diminuir a ocorrência de conflitos (restrição forte). O teste dos métodos em duas instituições encontrou soluções consideravelmente melhores do que as soluções encontradas manualmente. Mais especificamente tratando de problemas de programação de horários em escolas, algoritmos baseados em busca tabu foram propostos em COSTA (1994), SCHAERF (1999) e ALVAREZ-VALDÉS et al. (1996). Nos três trabalhos, foram analisadas instâncias de problemas. Na busca tabu de COSTA (1994) a vizinhança é definida pela mudança no período de alocação de uma determinada aula. A memória de curto prazo é implementada através de duas listas tabu: Essas duas listas consideram, respectivamente, a movimentação de

uma aula e a movimentação de uma aula para um determinado período, sendo que a primeira lista é menor que a segunda. O critério de aspiração considera alterações não somente no valor da função objetivo, mas em componentes desta. A diversificação é implementada através da oscilação dos pesos na função objetivo relacionados à ocorrência de conflitos: faz-se uma redução periódica desses pesos permitindo que o processo de busca visite outras regiões do espaço de busca. No método de SCHAERF (1999) os movimentos podem envolver a mudança de período de uma aula na agenda de um professor, mas também a troca de períodos entre duas aulas. A busca tabu de Schaerf é realçada com componentes aleatórios: após certo número de iterações sem melhora é realizada a fase de RNA (*randomized non-ascendent method*), onde movimentos são sorteados e, caso não gerem conflitos, estes movimentos são realizados. Caso o movimento sorteado gere um conflito, considera-se a reparação desse movimento por meio da exploração (nesse caso sistemática) de um segundo movimento, gerando os chamados movimentos duplos. O método também considera a utilização de relaxação adaptativa como em GENDREAU et al. (1994).

O método híbrido baseado em Busca Tabu e GRASP (RESENDE; RIBEIRO, 2003), proposto por SOUZA (2000) e SOUZA et al. (2003), destacou-se dentre os algoritmos híbridos estudados em SOUZA (2000), onde são apresentadas e avaliadas computacionalmente várias propostas de heurísticas para o PPTC, incluindo Simulated Annealing, Busca Tabu e Otimização Microcanônica (TORREÃO; ROE, 1995). Os algoritmos propostos agregavam um método de melhoramento baseado em caminhos mínimos (SOUZA et al., 2000) para a pesquisa de movimentos compostos, o qual é utilizado tanto para acelerar a obtenção de soluções factíveis quanto para o melhoramento dessas. A diversificação é implementada através da reinicialização com a fase de construção da meta-heurística GRASP. Nesse trabalho, o método híbrido baseado em Busca Tabu superou consideravelmente os métodos baseados em *Simulated Annealing* e Otimização Microcanônica.

O método *Simulated Annealing* foi apresentado no contexto de otimização em KIRKPATRICK et al. (1983). Baseado em busca local, assim como busca tabu, também considera a passagem da solução corrente para uma solução vizinha de pior qualidade na função objetivo. Nesse caso, no entanto, um componente aleatório, ao invés da utilização de memória, é considerado para decidir a aceitação do movimento. A probabilidade de aceitação de movimentos desse tipo diminui gradualmente no decorrer da busca. O trabalho de ABRAMSON (1991) realiza experimentos com versões sequenciais e paralelas do *Simulated Annealing*. O conjunto de problemas teste consistia em problemas artificiais, gerados de modo que a solução ótima fosse conhecida (sem conflitos, visto que havia sido a única restrição considerada) e um problema real, com algumas restrições adicionais. A versão paralela do método de *Simulated Annealing*, que consistia em uma implementação com granularidade fina (CUNG, 2002) é implementada em um sistema paralelo com memória compartilhada, apresentando uma aceleração considerável.

Vários algoritmos genéticos foram propostos para a resolução de problemas de programação de horários. O trabalho de ABRAMSON et al. (1992) propuseram um algoritmo genético paralelo para a resolução de problemas de programação de horários em escolas. No problema considerado, somente leva-se em conta a não existência de conflitos. Experimentos com instâncias artificiais são apresentados. O algoritmo genético híbrido de COLORNI et al. (1998) considera um problema com mais requisições reais. Nesse algoritmo é utilizada uma combinação de operadores genéticos que consideram características do problema e uma busca local em múltiplas vizinhan-

ças. Comparações com implementações bastante simples de *Simulated Annealing* e Busca Tabu são apresentadas, sendo que os melhores resultados foram apresentados pelo algoritmo genético híbrido e pela busca tabu. Um algoritmo genético híbrido também é apresentado em STEFANO et al. (2001).

Técnicas de Programação Linear, incluindo Programação Linear Inteira, também foram utilizadas para a geração de quadros de horários. Na maioria dos casos, busca-se a produção de quadros de horários provavelmente ótimos. Considerando que a prova da otimalidade pode implicar a resolução de um número computacionalmente intratável de problemas lineares, trabalhos iniciais como em LAWRIE (1969) tratavam apenas parte do problema hoje conhecido como o problema de programação de horários. Em LAWRIE (1969), além de o problema considerar apenas a não ocorrência de conflitos, cabia ao usuário do software informar previamente quais cursos podiam ser oferecidos concomitantemente, através da criação de “arranjos”. Em DREXL et al. (1997) a formulação compacta apresentada com variáveis binárias, permitiu somente a resolução de instâncias artificiais de menor dimensão, considerando o tamanho de problemas reais típicos.

Posteriormente, progressos na capacidade de resolução de problemas de Programação Linear Inteira Mista (PLIM) trouxeram um renovado interesse na aplicação de PLIM para a resolução de problemas de programação de horários e permitiram que problemas reais com importantes considerações práticas fossem tratados em AVELLA et al. (2005), DASKALAKI (2004) e BIRBAS et al. (1997). Uma particularidade dos modelos utilizados em AVELLA et al. (2005) e BIRBAS et al. (1997), bastante relacionados com o PPTC, é o fato de que a questão da compatibilidade de horários para professores não é avaliada na função objetivo. Em BIRBAS et al. (1997), por exemplo, especifica-se um conjunto de dias de folga para cada professor, já em AVELLA et al. (2005) especifica-se um número máximo de dias em que o professor deve comparecer à instituição de ensino. Essa abordagem, apesar de possibilitar uma considerável diminuição na dimensão do espaço de busca, traz a desvantagem de que na especificação do número de dias de folga/trabalho de cada professor devem ser feitas escolhas sábias: a requisição de horários tão compactos quanto possível, considerando cada professor isoladamente pode resultar em um problema sem solução viável, que somente poderá tornar-se viável através da relaxação desse requisito para alguns professores. A alteração manual do modelo para permitir a factibilidade e oferecer quadros de horários compactos para professores pode excluir soluções ótimas quanto a esse requisito.

Em PAPOUTSIS et al. (2003) é apresentada uma aplicação de Geração de Colunas de DANTZIG; WOLFE (1960) para um modelo com um número exponencial de variáveis, para uma variante do PPT encontrada na Grécia. Nesse modelo, cada variável binária representa um possível quadro de horários semanal de um professor, sendo que a maioria das restrições é considerada na geração das colunas, ou seja, tem-se quase que exclusivamente restrições fortes, de modo que a função objetivo tem poucos (dois) componentes. A determinação de colunas atrativas é realizada através da busca no espaço de soluções factíveis com a estratégia LIFO (o mais profundo deve ser avaliado primeiro). O procedimento procede iterativamente com a adição de novas colunas, enquanto alguma melhora significativa na função objetivo é observada. Uma vez que a solução resultante do procedimento apresentado geralmente não é inteira, os autores desenvolveram um procedimento específico que tenta encontrar a integralidade através da fixação de alguns quadros de horários de professores. Deve-se observar que o procedimento por esses autores é heurístico, sem garantia de produção da solução ótima.

Apesar da existência de várias publicações que tratam de problemas iguais ou semelhantes, poucos progressos foram realizados no campo de análises comparativas entre as implementações dos diferentes métodos, com eventuais exceções para casos em que os autores implementem as alternativas concorrentes do início como em COLORNI et al. (1998) e SOUZA (2000). No trabalho de ABRAMSON (1991), a comparação com os resultados de GANS (1981) é realizada utilizando-se instâncias diferentes das do trabalho anterior, visto que o autor não teve acesso aos dados completos que definiam as instâncias. Instâncias de mesma dimensão e conteúdos aleatórios foram utilizadas para os experimentos comparativos. O autor afirma em ABRAMSON (1991) que os resultados desse experimento não permitem nenhuma comparação absoluta entre os trabalhos. SCHAERF (1999) argumenta que suas escolhas com relação à representação de solução e definição da vizinhança são mais apropriadas do que as utilizadas em COLORNI et al. (1998) e COSTA (1994), respectivamente, sem apresentar resultados experimentais comparativos que justifiquem essa afirmação. De fato, nenhum dos trabalhos considerou o mesmo conjunto de instâncias, o que torna difícil a elaboração de conclusões sobre quais escolhas foram as mais apropriadas. Nos três trabalhos, a avaliação dos resultados é realizada considerando comparações com soluções manuais, um critério importante mas nem sempre preciso.

As pesquisas relacionadas com o problema de otimização de grade horária continuam muito ativas. Algumas referências muito importantes de publicação recente podem ser de grande interesse. Em AKKAN et al. (2018) é usado um algoritmo genético híbrido multiobjetivo onde além da função objetivo tradicional é incorporada uma medida da robustez da solução encontrada. Em SORIA-ALCARAZ et al. (2018) é usado o algoritmo de busca iterada (*Iterate Local Search*) que permite encontrar soluções de excelente qualidade. Em BABAEI et al. (2015) é realizada uma revisão bibliográfica das técnicas de otimização usadas no problema de otimização de grade horária disponíveis na literatura especializada. Em BAGGER et al. (2019) é usada uma técnica de geração de colunas para otimizar o problema de grade horária e encontra soluções ótimas de uma parcela dos problemas que são testados do tipo de problema analisado na pesquisa. Em FONSECA et al. (2017) é desenvolvido um modelo de programação inteira para um tipo de problema de grade horária e que apresenta excelentes resultados. Em SAVINIEC et al. (2018) aparece a proposta de formular a paralelização de meta-heurísticas para resolver o problema de otimização de grade horária. A proposta mais promissora é aquela que usa a paralelização de um algoritmo de busca local iterada que incorpora a estratégia de diversificação e intensificação usando informação baseada em memória. A proposta de otimização formulada apresenta excelente desempenho.

Outros modelos de otimização matemática para diferentes tipos de formulações do problema de otimização de grade horária podem ser encontrados em AL-YAKOOB et al. (2006); AL-YAKOOB et al. (2007), DASKALAKI et al. (2004); DASKALAKI et al. (2005). Outras aplicações de heurísticas e meta-heurísticas aplicadas ao problema de programação ótima de horários podem ser encontrados em ALVAREZ-VALDES et al. (2002); BURKE (2010); DE CAUSMAECKER et al. (2009); HERTZ et al. (1987); KAHAR et al. (2010); LU et al. (2010); MEISELS et al. (2004); NONOBE et al. (1998); QU et al. (2009); ZHANG D. et al. (2010). Artigos publicados em eventos especializados do problema de grade horária podem ser encontrados em BURKE et al. (1997); BURKE et al. (2000); BURKE et al. (1996); CARTER et al. (2001); COOPER et al. (2001). Tópicos que avaliam e apresentam uma informação mais geral sobre o problema de otimização de grade horária podem ser encontrados em BRILFORD et al. (1999); BURKE et al. (1997); BURKE et al. (2007); BURKE et al. (2002); BURKE et al. (2004);

DIMOPOULOU et al. (2001); KANG et al. (1997); SANTOS (2007); VANDEN BROEK et al. (1985) e WERRA et al. (1997).

Informação do site da Competição Internacional da Otimização do Problema de Grade horária se encontra na referência (INTERNATIONAL TIMETABLING COMPETITION).

Capítulo 3

Introdução sobre as Meta-heurísticas

Neste Capítulo é apresentada uma análise sobre as heurísticas e as meta-heurísticas priorizando o aspecto conceitual e a forma de aplicação. Assim, inicialmente, apresenta-se os conceitos básicos que devem ser levados em conta na aplicação de heurísticas e meta-heurísticas na solução de problemas complexos. A seguir são apresentados de forma resumida os aspectos mais importantes relacionados com cada heurística e meta-heurística específica. A análise detalhada da meta-heurística VNS, usada neste trabalho é abordada no próximo capítulo.

3.1 Aspectos Fundamentais Relacionados com as Heurísticas e as Meta-heurísticas

As heurísticas e as meta-heurísticas são técnicas de otimização alternativas das técnicas clássicas de otimização. Assim, as heurísticas e as meta-heurísticas são usadas para resolver problemas de otimização e são concorrentes com as diferentes técnicas clássicas de otimização tais como as técnicas que resolvem problemas de programação linear, programação não linear, programação linear inteira mista e programação não linear inteira mista. Para resolver um problema de otimização usando técnicas clássicas, o processo de otimização geralmente tem dois passos consecutivos, isto é, inicialmente se desenvolve o modelo matemático a ser otimizado e depois se escolhe a técnica de otimização. Na formulação do modelo matemático geralmente a decisão mais difícil é escolher de forma adequada as variáveis de decisão do problema que permite formular a função objetivo e as restrições do modelo matemático.

Quando se resolve um problema de otimização usando as heurísticas e as meta-heurísticas, a existência de um modelo matemático para o problema a ser resolvido pode ser de interesse secundário ou irrelevante. Em outras palavras, uma meta-heurística pode resolver um problema de otimização sem a necessidade de dispor do modelo matemático para esse problema. Dessa forma, usar uma meta-heurística pode ser adequado e oportuno para resolver um problema de otimização quando o problema a ser otimizado não tem modelo matemático (porque ainda não foi descoberto ou de fato não é possível formular um modelo matemático para esse problema) ou quando existe um modelo matemático para o problema a ser resolvido, mas as técnicas de otimização e os *solvers* correspondentes disponíveis não conseguem resolver o modelo matemático, para determi-

nadas instâncias do problema, porque aparecem problemas de memória para armazenamento de informação ou porque o tempo necessário para convergência é muito grande. Adicionalmente, as meta-heurísticas podem ser muito importantes para resolver problemas que exigem uma solução em tempos de processamento pequenos (segundos ou minutos) o que acontece com as aplicações em tempo real ou em programação da operação de curto prazo. Assim como o aspecto mais importante na formulação de um modelo matemático é a escolha adequada e eficiente das variáveis de decisão, na aplicação das meta-heurísticas a decisão que pode ser considerada mais importante é a escolha adequada e eficiente de uma forma de representar uma proposta de solução para o problema a ser otimizado.

Na utilização das meta-heurísticas em geral e, particularmente da meta-heurística VNS, deve-se especificar um conjunto de características relacionadas com a sua aplicação na resolução de um tipo de problema complexo. Essas características são as seguintes: (1) uma forma de representação de uma proposta de solução que deve identificar de forma adequada um elemento do espaço de busca, (2) uma forma adequada de identificação da qualidade da solução encontrada, (3) uma forma de identificar se uma proposta de solução é factível ou infactível, (4) uma estratégia de decisão em relação às soluções infactíveis, (5) uma estratégia para encontrar a primeira proposta de solução (ou o primeiro conjunto de soluções) que se transforma na solução corrente, (6) uma forma de caracterização de vizinhança e (7) uma estratégia de escolha do vizinho mais adequado para realizar a transição, isto é, o vizinho que deve ser escolhido como a nova solução corrente. A diferença fundamental entre as meta-heurísticas está nesta última característica.

De todas as características mencionadas anteriormente, a mais importante é a escolha de uma forma de representação de um elemento do espaço de busca. Essa proposta condiciona as outras características do problema e a qualidade da estratégia geral de otimização. A forma de representação de uma proposta de solução deve permitir encontrar o valor da função objetivo, ou seu equivalente, dessa proposta de solução, assim como verificar se a proposta é factível ou infactível. Também a forma de representação de uma proposta de solução permite especificar a estrutura de vizinhança para realizar as transições através do espaço de busca do problema.

3.2 Revisão sobre as Heurísticas

As heurísticas são técnicas de otimização que geralmente encontram soluções de boa qualidade de problemas complexos em um tempo de processamento muito pequeno. Deve-se observar que entre as décadas de 1960 e 1970, as heurísticas foram as técnicas de otimização mais usadas e com maior sucesso para resolver problemas complexos do campo da otimização matemática, especialmente para aqueles problemas não lineares, discretos e não convexos.

A maioria das heurísticas encontram soluções de boa qualidade de problemas complexos em tempos computacionais relativamente rápidos. Adicionalmente, a maioria das heurísticas são simples de entender e também de implementar computacionalmente. Entretanto, as técnicas heurísticas renunciam, pelo menos do ponto de vista teórico, a encontrar a solução ótima global de um problema complexo. Em problemas de grande porte e complexos, as técnicas heurísticas raramente encontram as soluções ótimas. Um problema de otimização pode ser considerado como sendo complexo quando existe grande dificuldade para encontrar a solução ótima global devido principalmente a que o problema apresenta a característica da explosão combinatória quando o

tamanho do problema cresce e/ou porque o problema apresenta uma modelagem matemática complexa (variáveis inteiras ou discretas, função objetivo não linear e não diferenciável, restrições não lineares, região factível não convexa, etc.).

Uma técnica heurística pode ser muito simples como, por exemplo, o uso de bom senso ou a experiência de um especialista ou pode ser muito sofisticado, geralmente envolvendo a solução de modelos matemáticos relaxados em relação ao modelo original. É recomendável usar técnicas heurísticas de otimização nos seguintes casos (ROMERO; LAVORATO, 2012):

1. Quando não existe um método exato de otimização para resolver o problema em análise.
2. Quando a solução ótima não é muito importante do ponto de vista prático por diferentes motivos como, por exemplo, a existência de muitas soluções ótimas locais de qualidade muito próxima da solução ótima global.
3. Quando os dados usados apresentam incerteza elevada como acontece em muitos problemas relacionados com planejamento.
4. Quando existem limitações de tempo de processamento para encontrar a solução procurada e quando existem problemas de memória para armazenamento de dados. Problemas desse tipo podem aparecer em problemas de aplicação em tempo real.
5. Quando se pretende encontrar uma boa solução inicial para ser usada como ponto de partida na aplicação de uma técnica de otimização mais sofisticada como, por exemplo, quando se pretende usar uma meta-heurística sofisticada ou um algoritmo *branch and bound*.

Existem vários tipos de heurísticas na literatura especializada, mas duas delas apresentam interesse especial porque elas são usadas de diversas formas nas meta-heurísticas mais sofisticadas. Assim, apresenta-se de forma resumida o algoritmo heurístico construtivo (AHC) e a heurística de busca através de vizinhança (SDH, *Steepest Descent Heuristic*).

3.2.1 O Algoritmo Heurístico Construtivo

O algoritmo heurístico construtivo (AHC) é uma das técnicas heurísticas de otimização mais usadas para resolver problemas complexos e ainda é muito usado isoladamente ou integrado a meta-heurísticas mais sofisticadas. O mais popular dos algoritmos heurísticos construtivos é o tipo guloso (*greedy*).

O AHC é uma técnica de otimização que, em um processo passo a passo, gera uma solução geralmente de boa qualidade de um problema complexo. Em cada passo o AHC escolhe um elemento ou componente da solução que está sendo construída e no último passo termina de gerar uma solução válida. O elemento ou componente da solução que é escolhido em cada passo do AHC é identificado usando um indicador de sensibilidade que identifica o componente mais interessante a ser incorporada na solução em construção. Assim, a diferença fundamental entre

os AHCs usados para resolver um mesmo problema está no indicador de sensibilidade usado. Um AHC pode assumir a seguinte forma genérica (ROMERO; LAVORATO, 2012):

1. Armazenar os dados do problema e escolher o indicador de sensibilidade a ser usado. Escolher os componentes que podem ser incorporados na solução em construção (geralmente o processo é iniciado sem componentes).
2. Verificar se a solução em construção já representa uma solução válida. Caso seja factível então pare o processo porque foi encontrada a solução válida procurada. Em caso contrário ir ao passo 3.
3. Usando a solução em construção, resolver o problema que permite identificar o indicador de sensibilidade de todos os componentes do problema que ainda não foram incorporadas na solução em construção.
4. Usando a informação dos indicadores de sensibilidade encontrados no passo anterior identificar a componente que deve ser incorporada na solução em construção. Adicionar o componente identificado na solução em construção e voltar ao passo 2.

Para o problema de programação ótima de grade horária, podem-se desenvolver várias estratégias heurísticas construtivas para tentar encontrar uma proposta de solução que não viole as restrições obrigatórias.

3.2.2 O Algoritmo Heurístico de Busca Através de Vizinhança

O Algoritmo heurístico de busca através de vizinhança (SDH) é significativamente diferente do algoritmo heurístico construtivo do tipo guloso. No AHC se gera apenas uma solução válida através de uma sequência de passos e usando um indicador de sensibilidade. No algoritmo SDH, o processo é geralmente iniciado a partir de uma solução factível e na sequência são encontradas novas soluções factíveis percorrendo o espaço de busca e passando sempre para a melhor solução vizinha. A solução inicial que o algoritmo SDH precisa pode ser gerada de diversas formas, mas a proposta mais interessante pode ser usar um AHC. Assim, o algoritmo heurístico construtivo e a heurística de busca através de vizinhança são complementares.

A estratégia fundamental da heurística SDH pode ser resumida da seguinte forma (ROMERO; LAVORATO, 2012):

- O processo de otimização é iniciado através de uma solução inicial que passa a ser chamada de solução corrente.
- Deve-se definir uma estrutura de vizinhança. Assim, deve existir uma forma de identificar as soluções que são consideradas vizinhas da solução corrente. As soluções vizinhas podem ser factíveis ou infactíveis.
- Na heurística SDH se passa da solução corrente para a melhor solução vizinha.

- O processo termina quando todas as soluções vizinhas são de pior qualidade que a solução corrente.

Como mencionado anteriormente sobre heurísticas e meta-heurísticas, para implementar a heurística SDH não é necessário usar o modelo matemático do problema em análise. Na verdade a heurística SDH pode resolver problemas de otimização que não tem modelagem matemática e essa característica torna a heurística SDH, assim como as meta-heurísticas, uma técnica de otimização relativamente distante da lógica de otimização usada na otimização clássica. Para uma melhor apresentação do SDH, apresenta-se a continuação de forma mais detalhada alguns tópicos comuns a um algoritmo SDH e às meta-heurísticas.

A *representação de uma proposta de solução* de um problema complexo representa a estratégia fundamental para entender o funcionamento da heurística SDH. Assim, a representação de uma proposta de solução é uma forma inequívoca de identificar um elemento do espaço de busca do problema de otimização e também para mostrar o resultado final do problema. O espaço de busca está formado por todas as propostas de solução que podem ser identificados usando a forma de representar uma proposta de solução escolhida para um determinado problema de otimização e usando as estratégias de vizinhança usadas para resolver o problema. Deve-se lembrar que um aspecto crucial na otimização clássica é a escolha adequada das variáveis de decisão. Na heurística SDH um aspecto crucial e fundamental é a escolha de uma forma de representar uma proposta de solução. A forma de representar uma proposta de solução na heurística SDH substitui as variáveis de decisão no problema de otimização clássica.

Uma proposta de solução está *adequadamente representada* quando, a partir dessa informação, é possível encontrar o valor da função objetivo (ou seu equivalente) do problema e determinar se a proposta é factível ou infactível. Uma proposta de solução está *eficientemente representada* para sua utilização em uma heurística SDH ou em uma meta-heurística quando os operadores desse algoritmo podem ser eficientemente implementados. Deve-se observar que podem existir várias formas de representar uma proposta de solução (ROMERO; LAVORATO, 2012).

Para algumas meta-heurísticas como, por exemplo, nos algoritmos genéticos e evolutivos em geral a proposta de representar uma proposta de solução deve permitir uma implementação eficiente dos operadores existentes na meta-heurística. No caso do algoritmo genético, a representação de uma proposta de solução eficiente deve permitir implementar de forma eficiente os operadores de seleção, recombinação e mutação. Portanto, a forma de representação de uma proposta de solução escolhida para resolver um problema complexo usando a heurística SDH ou uma meta-heurística deve permitir identificar de maneira única um elemento do espaço de busca, deve permitir encontrar o valor da função objetivo ou equivalente e deve permitir verificar se a proposta de solução é factível ou infactível. Adicionalmente, a proposta de codificação deve permitir definir e implementar de forma adequada as estruturas de vizinhança e/ou implementar de forma eficiente os operadores existentes na meta-heurística (ROMERO; LAVORATO, 2012).

A heurística SDH realiza a busca através de vizinhança, isto é, a partir da solução corrente, passa-se para a melhor solução vizinha. Assim, deve-se definir a vizinhança da solução corrente. Seja p_k a codificação da solução corrente de um problema complexo que está sendo resolvido pela heurística SDH. Uma *solução vizinha* de p_k é qualquer proposta de solução que pode ser

encontrada com a implementação de um mecanismo de perturbação da solução corrente. As soluções vizinhas são armazenadas no conjunto $N_k(p)$. O número de soluções vizinhas de p_k é conhecida como a cardinalidade de $N_k(p)$ e denotada através da notação $|N_k(p)|$. Assim, para definir a vizinhança da solução corrente apenas é necessário definir o mecanismo de perturbação.

Supor que temos disponível a solução corrente p_k e se conhece o mecanismo de perturbação para encontrar os elementos de $N_k(p)$. Nesse contexto, os elementos de $N_k(p)$ podem representar soluções factíveis ou infactíveis para o problema que está sendo resolvido. A ideia central relacionada com a definição de $N_k(p)$ deveria ser idealizar uma vizinhança formada apenas por elementos factíveis. Entretanto, esse tipo de proposta nem sempre é possível e depende de muitos fatores tais como a forma de representar uma proposta de solução, o tipo de problema que está sendo resolvido e da forma de definição de vizinhança. Quando não existe possibilidade de gerar apenas soluções vizinhas então a heurística SDH deve decidir a forma de tratamento das soluções vizinhas. Na maioria das aplicações as propostas de solução infactíveis são descartadas, mas em outras aplicações podem ser consideradas para análise e avaliadas considerando, por exemplo, uma penalização dessas soluções na função objetivo. Em resumo, as diferentes formas de definir a vizinhança da solução corrente diferem apenas no mecanismo de perturbação usado. Para cada tipo de problema é possível definir vários tipos de vizinhança. Assim, a definição da vizinhança, junto com a proposta de representação de uma proposta de solução, é um dos tópicos cruciais na formulação de uma heurística SDH eficiente.

Após apresentar com certo detalhe a terminologia usada na heurística SDH, que é praticamente a mesma usada na implementação das meta-heurística, pode-se apresentar a forma genérica da heurística SDH. Assim, a forma genérica da heurística SDH assume a seguinte forma (ROMERO; LAVORATO, 2012):

1. Passo preliminar: Montar os dados do problema. Escolher uma forma de representar uma proposta de solução denominada de p . Identificar uma forma de avaliar a qualidade da função objetivo ou equivalente e denominada $f(p)$. Definir a estrutura de vizinhança a ser usada o que caracteriza o espaço de busca.
2. Encontrar uma solução inicial p_o que se transforma na solução corrente p_c .
3. Identificar e avaliar todas as soluções vizinhas da solução corrente p_c e identificar a melhor solução vizinha p^{best} .
4. Se $f(p^{best}) < f(p)$ então a melhor solução vizinha é melhor que a solução corrente e, portanto, fazer $p_c = p^{best}$ e voltar ao passo 3. Em caso contrário pare o processo e a solução encontrada pela heurística SDH é p_c (geralmente um ótimo local).

Em relação com a heurística SDH, deve-se realizar as seguintes observações (ROMERO; LAVORATO, 2012):

- O passo 1 é muito mais importante do que normalmente é considerado por muitos pesquisadores. A escolha de uma forma de representar uma proposta de solução de forma eficiente é crucial, assim como a caracterização da estrutura de vizinhança. Nesse passo também deve

ser decidido como devem ser tratadas as propostas de solução vizinhas que são inactíveis. Assim, é de responsabilidade do pesquisador escolher ou definir a forma de representar uma proposta de solução a ser usada, do tipo de vizinhança escolhido e a forma de tratar as soluções vizinhas inactíveis.

- No passo 2 a heurística SDH exige apenas uma solução inicial. Assim, essa solução inicial pode ser encontrada de forma trivial escolhendo, por exemplo, uma solução gerada de forma aleatória ou pode ser gerada de forma sofisticada usando, por exemplo, um AHC conhecido e eficiente para o tipo de problema em análise (um ótimo local). Deve-se observar que em muitos problemas pode ser muito difícil encontrar uma solução inicial factível como acontece no problema de otimização de grade horária abordado neste trabalho. Portanto, em geral, a solução inicial p_o pode ser factível ou inactível. Caso seja inactível, então deve-se mudar a estratégia de qualidade para avaliar as soluções vizinhas. A estratégia mais popular consiste em penalizar a função objetivo das propostas de solução inactíveis. Também, no passo 2 é exigido apenas uma solução inicial.
- No passo 3, a heurística SDH exige avaliar a qualidade de todas as soluções vizinhas e identificar a melhor solução vizinha. Esse passo pode exigir muito tempo de processamento, especialmente em determinados problemas em que avaliar a qualidade de uma solução vizinha pode exigir resolver um problema relativamente complexo como, por exemplo, um problema de programação linear ou não linear. Mesmo que a avaliação da qualidade de uma solução vizinha seja relativamente rápida, o passo 3 pode exigir muito tempo de processamento para realizar uma transição da solução corrente para a nova solução corrente (melhor vizinho).
- No passo 4 a heurística SDH termina o processo se a melhor solução vizinha for de pior qualidade que a solução corrente. Assim, se a vizinhança for definida de forma inadequada então a heurística SDH pode terminar encontrando uma solução ótima local de pobre qualidade. Também, devemos observar que quando a heurística SDH converge, então a solução corrente é também a incumbente. Essa característica não acontece na maioria das meta-heurísticas, exceto na meta-heurística de busca em vizinhança variável (VNS, do inglês *Variable Neighborhood Search*) usado neste trabalho.

A abordagem usada para analisar o funcionamento da heurística SDH é usada a seguir para apresentar as principais meta-heurísticas como sendo uma sofisticação da heurística SDH. Assim, toda a terminologia usada para explicar o funcionamento da heurística SDH pode ser usada para explicar o funcionamento da heurística SDH. Finalmente, pode ser oportuno mencionar as limitações da heurística SDH. Assim, as principais limitações da heurística SDH, que de alguma forma são contornadas pelas meta-heurísticas, são as seguintes:

1. A heurística SDH inicia o processo de otimização através de espaço de busca a partir de uma única proposta de solução inicial (a partir de um elemento do espaço de busca). Assim, existe a possibilidade de que o SDH procure apenas em uma região pequena do espaço de busca e termine em uma solução ótima local. Em outras palavras, o SDH não tem capacidade plena para percorrer uma grande parcela do espaço de busca. Nesse contexto, propostas de otimização que iniciam o processo de otimização através de um conjunto inicial de propostas de solução tem maior capacidade de percorrer maiores regiões do espaço de busca.

2. A heurística SDH identifica, avalia e ordena todas as propostas de solução vizinhas. O custo de tempo de processamento para realizar essa tarefa pode ser muito grande. Deve-se lembrar que todo esse esforço de processamento é apenas para realizar uma transição, isto é, passar da solução corrente para o melhor vizinho, que se transforma na nova solução corrente. Nesse contexto, a proposta de realizar a transição após identificar a primeira solução vizinha que seja melhor que a solução corrente pode acelerar o processo de busca através do espaço de busca. Outra alternativa eficiente pode ser implementar uma estratégia de redução da vizinhança de forma que a melhor solução vizinha seja a mesma na vizinhança completa e na vizinhança reduzida.
3. O critério de parada da heurística SDH é um assunto crítico. O SDH termina o processo de otimização quando a melhor solução vizinha é pior que a solução corrente e esse requisito pode ser cumprido rapidamente e terminando o processo de otimização em um ótimo local que não seja de boa qualidade.
4. A forma de escolha da proposta de solução inicial para iniciar o processo de busca também é um assunto relevante a ser analisado.

3.3 Revisão sobre as Meta-heurísticas

As meta-heurísticas, segundo GLOVER (1994) são técnicas de otimização que gerenciam uma interação entre as estratégias de busca local e as estratégias de nível superior para criar um processo de otimização com capacidade de sair de soluções ótimas locais e realizar uma busca robusta através de espaço de busca. Alternativamente, pode-se definir uma meta-heurística como sendo um processo de otimização representado por uma generalização e/ou integração do algoritmo heurístico construtivo de tipo guloso e a heurística de busca através de vizinhança de forma que seja possível encontrar soluções de qualidade percorrendo de forma eficiente o espaço de busca. Em outras, palavras, uma meta-heurística pode ser interpretada como uma generalização e/ou integração do AHC e da heurística SDH mostrados anteriormente.

Nesta Seção, apresenta-se as principais meta-heurísticas usadas no campo da pesquisa operacional. Nessa apresentação um tanto resumida, apresenta-se a meta-heurística informando de forma rápida a origem e tentando apresentar essas meta-heurísticas como sendo uma generalização e/ou integração do AHC e da heurística SDH.

3.3.1 Simulated Annealing

Simulated Annealing (SA) é uma das primeiras meta-heurísticas que foi usado com muito sucesso na otimização de problemas complexos na pesquisa operacional. SA foi inventado após verificar que existiam muitas semelhanças entre a técnica de construção de cristais perfeitos usando a técnica de *annealing* e a otimização de um problema complexo no campo da pesquisa operacional.

Existem muitas técnicas usadas na construção de cristais perfeitos e uma delas é a técnica

de *annealing*. A técnica de *annealing* consiste em esquentar um material até temperaturas elevadas na qual existe muita movimentação molecular do material esquentado e, portanto, um novo arranjo dos átomos do material. A partir desse estado, se o material for esfriado lentamente, então a movimentação molecular também diminui. Se esse processo de diminuição for adequadamente controlado preservando o chamado quase equilíbrio termodinâmico na qual a temperatura deve ser diminuída lentamente, então existe grande possibilidade de que o material se transforme em um cristal perfeito.

Resumindo, usando as semelhanças que existem entre a técnica de *annealing* na construção de cristais perfeitos e na otimização de problemas complexos no campo da pesquisa operacional, foi desenvolvido o algoritmo de *Simulated Annealing* que na formulação básica assume a seguinte forma (problema de minimização) (ROMERO; LAVORATO, 2012):

1. Passo preliminar: Montar os dados do problema. Escolher uma forma de representar uma proposta de solução denominada de p . Identificar uma forma de avaliar a qualidade da função objetivo ou equivalente e denominada $f(p)$. Definir a estrutura de vizinhança a ser usada o que caracteriza o espaço de busca. Escolher os parâmetros de SA tais como o parâmetro chamado de temperatura inicial T_o , o critério de parada, o número de tentativas de transição no primeiro nível de temperatura N_o , o parâmetro ρ que controla o número de tentativas de transição em cada nível de temperatura e o parâmetro β que controla a diminuição do parâmetro temperatura.
2. Encontrar uma solução inicial p_o que se transforma na solução corrente p_c e fazer $N_k = N_o$, $s = 0$.
3. Identificar e avaliar uma solução vizinha p_v escolhida aleatoriamente.
4. Se $f(p_v) < f(p_c)$ então a solução vizinha é melhor que a solução corrente e, deve-se realizar a transição, isto é, $p_c = p_v$ e ir ao passo 4. Em caso contrário gere um número aleatório entre 0 e 1, $P(0, 1) = random[0, 1]$, e seja $\Delta f(p) = f(p_v) - f(p_c)$. Assim, se $exp[-\frac{\Delta f(p)}{T_k}] > P(0, 1)$ então, deve-se realizar a transição e $p_c = p_v$. Em caso contrário, a solução corrente é preservada. Ir ao passo 4.
5. $s = s + 1$. Se $s < N_k$ então ir ao passo 3. Em caso contrário ir ao passo 6.
6. Se o critério de parada foi cumprido então pare. Em caso contrário, fazer $T_{k+1} = \beta T_k$ e $N_{k+1} = \rho N_k$. $k = k + 1$. Voltar ao passo 3.

Em relação com o algoritmo SA, deve-se realizar as seguintes observações:

- A relação $exp[-\frac{\Delta f(p)}{T_k}]$ sempre é positiva e varia entre 0 e 1. Por esse motivo faz sentido a comparação com um número aleatório $P(0, 1)$ que também varia entre 0 e 1. Adicionalmente, se T_k é muito grande ou $\Delta f(p)$ é muito pequeno então $exp[-\frac{\Delta f(p)}{T_k}]$ assume um valor próximo de 1. Por outro lado, se T_k é muito pequeno ou $\Delta f(p)$ é muito grande, então $exp[-\frac{\Delta f(p)}{T_k}]$ assume um valor próximo de zero.

- Da análise anterior podemos concluir que se a solução vizinha tem um valor da função objetivo que é de pior qualidade que da solução corrente mas de valor muito próxima e/ou se o processo se encontra nas fases iniciais (o valor do parâmetro de temperatura elevada), então existe uma probabilidade elevada de aceitar uma solução vizinha de pior qualidade. Por outro lado, se o valor da função objetivo da solução vizinha é de pior qualidade e muito distante do valor da função objetivo da solução corrente e/ou o processo se encontra nas fases finais (valor do parâmetro temperatura baixa) então a probabilidade de aceitar uma solução vizinha de pior qualidade é pequena. Em resumo, nas fases iniciais do processo podem ser aceitas soluções vizinhas de pior qualidade com mais facilidade (priorizando um processo de diversificação) e nas fases finais do processo raramente são aceitas soluções vizinhas de pior qualidade (priorizando o processo de intensificação). Finalmente, em qualquer estágio do processo, soluções vizinhas ligeiramente piores que a solução corrente têm maior probabilidade de serem aceitas comparadas com soluções vizinhas piores e muito distantes da solução corrente.
- A relação $\exp\left[\frac{-\Delta f(p)}{T_k}\right]$ pode ser substituída por qualquer outra com a única condição de que desempenhe um papel equivalente, isto é, priorize as soluções vizinhas de pior qualidade da forma mostrada nos passos anteriores.

Deve-se observar que o algoritmo SA básico apresentado anteriormente pode ser obtido apenas como uma generalização da heurística SDH. Comparando a heurística SDH e o algoritmo SA podemos verificar as seguintes diferenças fundamentais:

- A heurística SDH avalia todas as soluções vizinhas para identificar o vizinho de melhor qualidade. Assim, o passo 3 da heurística SDH é muito demorada. Por outro lado, SA escolhe aleatoriamente uma solução vizinha e decide se realiza a transição sem conhecer nem avaliar as outras soluções. Nesse contexto, nas fases iniciais SA realiza transições com maior intensidade que a heurística SDH, mas nas fases finais do processo ambas técnicas de otimização devem apresentar comportamento muito próximas.
- SA pode sair de um ótimo local ao aceitar de forma probabilística transições para soluções vizinhas de pior qualidade. Essa é a principal diferença entre a heurística SDH e a meta-heurística SA.
- Obviamente SA apresenta um maior tempo de processamento e deve encontrar soluções de melhor qualidade que a heurística SDH.

3.3.2 Tabu Search - Busca Tabu

Tabu Search (TS) é uma meta-heurística inventada por GLOVER (1986) e podem ser encontradas informações importantes relacionadas com o *Tabu Search* em GLOVER et al. (1997) e GLOVER et al. (2003). Ao contrário da maioria das meta-heurísticas que usaram comportamentos ou características existentes em ramos do conhecimento distantes da otimização matemática,

Glover usou apenas conhecimento existente no campo da otimização matemática. Antes de inventar o TS, Glover já era um pesquisador destacado em otimização clássica, especialmente nas técnicas de otimização de problemas de programação inteira (ROMERO; LAVORATO, 2012).

A ideia central de Glover é mostrar de que é possível inventar uma estratégia de busca inteligente percorrendo o espaço de busca de forma eficiente e seletiva. Nesse processo é fundamental integrar no processo de busca as estratégias de intensificação e de diversificação. Nesse contexto, intensificar significa realizar uma busca mais intensa em torno da solução corrente, por exemplo, aumentando o tamanho da vizinhança ou melhorando a qualidade da vizinhança. Por outro lado, diversificar significa sair de uma região do espaço de busca e, de forma proposital, atingir uma região distante para novamente realizar algum processo de intensificação.

A formulação básica de TS que está inspirado fundamentalmente no uso da estratégia de intensificação assume a seguinte forma:

1. Passo preliminar: Montar os dados do problema. Escolher uma forma de representar uma proposta de solução denominada de p . Identificar uma forma de avaliar a qualidade da função objetivo ou equivalente e denominada $f(p)$. Definir a estrutura de vizinhança a ser usada o que caracteriza o espaço de busca. Identificar os atributos que devem ser proibidos e o critério de aspiração. Escolher os parâmetros do algoritmo tais como a duração da lista tabu. Escolher o critério de parada.
2. Encontrar uma solução inicial p_o que se transforma na solução corrente p_c .
3. Identificar e avaliar todas as soluções vizinhas da solução corrente p_c , ordenar essas soluções vizinhas por qualidade sendo que o primeiro da lista é a melhor solução vizinha p^{best} .
4. Realizar a transição para a solução vizinha melhor classificada que não tem o atributo proibido ou se tem o atributo proibido, então satisfaz o critério de aspiração. Seja p_e a melhor solução vizinha escolhida, então fazer $p_c = p_e$.
5. Atualizar a incumbente e a lista de atributos proibidos. Se o critério de parada for satisfeito então pare. Em caso contrário voltar ao passo 3.

O algoritmo TS anteriormente apresentado é chamado de algoritmo TS básico que fundamentalmente usa memória de curto prazo, uma lista de atributos proibidos e um critério de aspiração, isto é, representa uma estratégia de otimização que prioriza a intensificação. Um atributo geralmente é representado pelo mecanismo que permite realizar a transição. Por exemplo, no problema de otimização de grade horária uma solução vizinha pode ser identificada pela troca de sala de aula de dois eventos no mesmo horário de programação. Nesse contexto a troca de sala de aula representa o atributo e proibir o atributo significa que o movimento reverso (os eventos trocam novamente de sala de aula voltando ao estágio anterior) fica proibido por um número especificado de transições. O critério de aspiração é uma medida da qualidade da função objetivo em relação com a função objetivo de uma referência. Assim, por exemplo, um atributo que se encontra proibido pode ser liberado para executar a transição se a função objetivo da solução vizinha for melhor que as encontradas em um número especificado de transições (por exemplo, 20 transições).

Técnicas de otimização tipo TS mais complexas podem ser implementadas onde o TS básico funciona como um módulo de otimização integrado em uma estrutura TS mais complexa. Essas estruturas mais complexas podem ser idealizadas usando outros operadores existentes em TS tais como a diversificação, a memória baseada em frequência, a lista de soluções de elite, o *path relinking*, entre outros. Para uma análise detalhada sobre a teoria e aplicação de TS ver GLOVER (1994). Em relação ao TS básico apresentado anteriormente, deve-se mencionar as seguintes observações (ROMERO; LAVORATO, 2012):

- A única diferença do algoritmo TS básico em relação com a heurística SDH é que TS realiza a transição para a melhor solução vizinha que não tem atributo proibido ou se tem o atributo proibido então cumpre com o critério de aspiração. Em outras palavras, se todas as soluções vizinhas da solução corrente são piores que a solução corrente, então o algoritmo TS passa para a menos pior, mas montando uma estratégia para tentar evitar visitar uma solução já visitada.
- Como o algoritmo TS pode passar para uma solução de pior qualidade e ao mesmo tempo apresenta uma estratégia gulosa, então deve existir uma estratégia para evitar voltar a visitar uma solução já visitada. Essa estratégia pode ser montada de forma explícita armazenando todas as soluções já visitadas (ou um conjunto menor representado pelas últimas soluções já visitadas) e verificando se a solução vizinha candidata a ser escolhida já foi visitada anteriormente ou pode ser armazenada de forma implícita armazenando apenas uma característica da solução visitada e chamada de atributo.
- O algoritmo TS básico não recomenda o armazenamento de soluções explícitas devido a problemas de memória para armazenamento da informação e pelo tempo de processamento necessário para verificar se uma solução vizinha já foi visitada anteriormente e, portanto, se encontra armazenada. Entretanto, essa premissa que era muito válida na década de 1980 pode não ter a mesma validade atualmente devido ao incremento elevado de disponibilidade de memória dos computadores modernos e também ao aumento da velocidade de processamento desses computadores. Assim, para determinadas aplicações pode ser oportuno avaliar a possibilidade do uso de memória explícita.
- Na estratégia implícita é usado o conceito de atributo. O atributo relacionado com uma proposta de solução é algum tipo de informação que permite identificar alguma característica da proposta de solução. O tipo de atributo mais usado é o próprio mecanismo usado na transição e que também permite a identificação de uma solução vizinha da solução corrente.
- A proibição através de atributos tenta evitar voltar a uma solução já visitada. Entretanto, como essa proibição termina após k_p transições previamente especificada, então é possível no futuro visitar uma solução já visitada e nesse caso acontece o fenômeno chamado de ciclagem e uma forma de atenuar esse problema é aumentando o valor de k_p . Por outro lado, todas as soluções vizinhas que compartilham um atributo proibido com uma solução já visitada também não poderiam ser visitadas. Nesse contexto, se o valor de k_p for muito grande, então podem existir muitas soluções vizinhas que não poderiam ser visitadas porque compartilham os mesmos atributos de soluções já visitadas e que ainda se encontram proibidas. Por esse motivo, o valor de k_p não pode ser muito grande. Adicionalmente, muitas soluções vizinhas que compartilham atributos proibidos com soluções já visitadas podem

ser de excelente qualidade e, eventualmente, a própria solução ótima global pode estar proibida. Para contornar esse problema foi inventada a estratégia de critério de aspiração.

- A ideia central do critério de aspiração é eliminar a proibição do atributo de uma solução vizinha de excelente qualidade porque compartilha o mesmo atributo de uma solução já visitada. Assim, se uma solução vizinha satisfaz um critério de qualidade que permite suspeitar que seja uma solução ainda não visitada, então eliminamos a proibição e realizamos a transição para essa solução vizinha. Esse critério de qualidade está relacionado com o valor da função objetivo. Um critério extremo, consiste em especificar que uma solução vizinha cumpre com o critério de aspiração, se for melhor que a incumbente. Nesse caso, existe a certeza absoluta de que essa solução vizinha nunca foi visitada anteriormente, mas apresenta a desvantagem de que raras vezes pode ser acionada o que pode ser quase equivalente a não ter critério de aspiração. Geralmente, pode ser mais oportuno escolher um critério de aspiração menos exigente. Assim, se a solução vizinha é melhor que as soluções visitadas nas últimas k_{ul} iterações então cumpre com o critério de aspiração.

Em resumo, a diferença fundamental entre um TS básico e uma heurística SDH está na escolha do vizinho para realiza a transição. O SDH termina o processo quando a melhor proposta de solução vizinha é pior que a solução corrente. Por outro lado, o TS realiza a transição para a melhor solução vizinha que não foi visitada anteriormente e, portanto, deve-se especificar um critério de parada de forma explícita.

3.3.3 O Algoritmo Genético

O algoritmo genético (AG) é uma meta-heurística inventada por Holland na década de 70 sendo que apenas na década de 80 teve sua aplicação de forma intensa para resolver problemas complexos no campo da pesquisa operacional. Para inventar o AG Holland encontrou semelhanças entre a forma de resolver um problema de otimização matemática e o processo de seleção natural e evolução das espécies. Na verdade, na natureza o processo de seleção natural e a evolução das espécies é a consequência de um processo de otimização estocástica que acontece em um determinado ambiente e em tempo real.

O processo de seleção natural e a evolução das espécies é um problema muito complexo para que seja imitado de forma adequada por um processo de otimização de um problema complexo do campo da pesquisa operacional. Assim, pode-se afirmar que o AG imita apenas uma parcela dos componentes que fazem parte do processo de seleção natural e de evolução das espécies. Algumas implementações do AG que não acompanham de forma adequada o processo de seleção natural e a evolução das espécies são as seguintes (ROMERO; LAVORATO, 2012):

- Um cromossomo na genética é considerado como equivalente a uma proposta de solução no problema de otimização. Na verdade uma proposta de solução deveria ser equivalente a um indivíduo. Adicionalmente, na genética, nas espécies mais complexas não existe um cromossomo, mas uma cadeia cromossômica.
- Um cromossomo na genética na verdade é um par cromossômico formado por duas cadeias de informação, um herdado do pai e outro herdado da mãe. Um elemento do cromossomo

é um gene que tem dois valores definidos (alelos) um em cada cadeia cromossômica. As duas informações representam uma unidade de informação genética. Na maioria dos casos existem apenas dois tipos de alelos (nessa posição genética e para todos os indivíduos da espécie) na posição correspondente a um gene. Assim, esse tipo de informação pode ser armazenado de forma binária (0 ou 1). Entretanto foi provado que para uma determinada espécie, existem vários tipo de alelo entre os elementos da espécie. Por exemplo, o gene que determina a cor dos olhos nos humanos. Portanto, a codificação binária usada nos AG não representa de forma adequada aquilo que acontece na natureza.

- A informação genética presente em um gene define o fenótipo, isto é, a característica externa do individuo. Assim, por exemplo, um tipo de alelo (informação genética) define o tipo de sangue humano (Rh^+ ou Rh^-). Assim, existiria uma relação biunívoca entre tipo de alelo e fenótipo, o que permite montar uma estratégia de codificação (tipo de alelo) e decodificação (fenótipo) de forma binária no AG. Entretanto, foi provado que em muitos casos apenas um tipo de alelo define várias características fenotípicas e também vários alelos correspondentes a vários genes definem apenas uma característica fenotípica. Adicionalmente, o fenômeno de dominância na genética em que um alelo domina o outro alelo é muito importante na genética e não se encontra representado no AG.
- O fenômeno de *crossing over* é fundamental para a existência da diversidade genética entre os indivíduos de uma espécie. A diversidade genética é a chave fundamental na seleção natural e na evolução das espécies. O fenômeno de *crossing over* acontece quando uma célula reprodutiva se duplica e se divide. Assim, uma célula reprodutiva (que é uma unidade celular com duas cadeias cromossômicas) após o *crossing over* gera quatro cadeias cromossômicas independentes. Acontece que na separação, uma cadeia cromossômica (por exemplo, um espermatozoide) é formada por parcelas da cadeia cromossômica herdadas do pai e da mãe, em um processo de recombinação bastante complexo. Em outras palavras, um espermatozoide é formado por pequenas parcelas da cadeia cromossômica do pai e da mãe, misturados por parcelas pequenas de cada um. Sem *crossing over*, um espermatozoide estaria formado apenas pela cadeia cromossômica herdada pelo pai ou pela mãe. Assim, um espermatozoide é uma meia célula que posteriormente pode se juntar a um óvulo para formar uma nova unidade celular (zigoto) e, portanto, um novo indivíduo. Assim, deve-se esclarecer que o fenômeno de *crossing over* e a reprodução sexuada acontecem em instantes muito diferentes.
- No AG o fenômeno de *crossing over* é imitado pelo operador de recombinação. Pode-se verificar que esse processo de imitação não é muito adequado.
- O fenômeno da mutação acontece na natureza quando a composição genética de um gene é alterado (no processo de duplicação e separação de uma célula reprodutiva). Geralmente, esse processo de mutação pode gerar um indivíduo de pior qualidade, mas às vezes pode gerar indivíduos de melhor qualidade. Esse fenômeno é adequadamente imitado no AG.

Em resumo, podemos afirmar que um AG imita de forma não adequada o processo de seleção natural e de evolução das espécies. Esse processo de imitação é mais crítico no fenômeno de *crossing over* e relativamente significativo na codificação binária. Entretanto, apesar de todas as observações realizadas anteriormente, o AG genético apresenta excelente desempenho na resolução de muitos problemas do campo da pesquisa operacional. Assim, sempre existiu essa

duvida entre os pesquisadores sobre esse aparente paradoxo, o que de alguma maneira explica a busca intensa de encontrar algoritmos inspirados na seleção natural e da evolução das espécies que incorporem de forma adequada o que acontece na natureza. Essa pode ser uma explicação para a grande diversidade de algoritmos desse tipo, que existe na literatura especializada e que podem ser denominados apenas como algoritmos evolutivos (ROMERO; LAVORATO, 2012).

Um algoritmo genético básico assume a seguinte forma:

1. Passo preliminar: Montar os dados do problema. Escolher uma forma de representação de uma proposta de solução denominada de p . Identificar uma forma de avaliar a qualidade da função objetivo ou equivalente e denominada $f(p)$. Escolher os parâmetros do algoritmo tais como o tamanho da população n_p , a taxa de recombinação ρ_r , a taxa de mutação ρ_m e o tipo de seleção. Escolher um critério de parada.
2. Gerar a população inicial, isto é, gerar um conjunto de n_p de propostas de solução que se transforma na população corrente.
3. Avaliar a qualidade $f(p)$ de todos os elementos da população e atualizar a incumbente, se possível.
4. Se o critério de parada for satisfeito, pare. Em caso contrário, ir ao passo 5.
5. Implementar o operador de seleção.
6. Implementar o operador de recombinação.
7. Implementar o operador de mutação, atualizar a população corrente e voltar ao passo 3.

Os primeiros algoritmos genéticos usavam a representação binária de uma proposta de solução. Essa forma de representar uma proposta de solução, de forma implícita, foi chamada de codificação. Assim, uma proposta de solução codificada na forma binária precisa de decodificação para identificar a proposta de solução original. Entretanto, os algoritmos genéticos modernos seguem a proposta de Michalewicz, sugerindo que uma proposta de solução deve ser representada seguindo a natureza e as características do problema. Assim, por exemplo, no problema de otimização de grade horária pode ser muito complexo representar uma proposta de solução de forma codificada na forma binária de representação. Entretanto, no mesmo problema pode ser idealizada a representação de uma proposta de solução de forma natural, isto é, na forma de uma representação matricial armazenando os eventos em uma grade horária.

A população inicial, isto é, o conjunto inicial de propostas de solução geralmente é montada de forma aleatória. Entretanto, em algoritmos genéticos mais sofisticados, o conjunto inicial de propostas de solução pode ser gerado usando algoritmos heurísticos construtivos eficientes de forma a gerar soluções de qualidade e diversificadas. O operador de seleção determina o número de participações que deve ter cada elemento da população (proposta de solução) na formação da nova população. Obviamente o número de participações deve ser um número inteiro. O operador de seleção é um dos operadores que foi mais discutido e analisado nas fases iniciais de aplicação dos algoritmos genéticos. Na proposta inicial do algoritmo genético, o operador de seleção era implementado usando a seleção proporcional. Entretanto, os algoritmos evolutivos

modernos preferem usar a seleção baseados em torneio devido aos problemas apresentados pela seleção proporcional. A seleção por torneio apresenta muitas vantagens tais como não precisa de padronização (o problema pode ser de maximização ou de minimização, assim como os valores da função objetivo podem ser positivos ou negativos), os valores da função objetivo não são muito determinantes, não apresenta perda de diversidade e não precisa da roleta.

A recombinação escolhe (geralmente de forma aleatória) duas propostas de solução com direito a gerar novos descendentes e recombina essas soluções gerando duas candidatas a novas soluções. Cada descendente está formado por parcelas de uma das duas soluções geradoras. Assim, usando a recombinação de um ponto, um descendente está formado por uma parcela continua de um gerador e a seguir outra parcela continua do outro gerador. A recombinação pode ser de um ponto, de dois pontos, vários pontos ou multiponto. Finalmente, a mutação significa uma pequena perturbação na composição de uma proposta de solução. Na representação binária, uma posição do vetor que tem valor atual de 0 deve ser trocada para o valor 1. Em outros tipos de representação, deve ser realizada uma pequena perturbação levando em conta o tipo de representação usado e as características do problema. Uma versão muito importante do algoritmo genético, que usa a lógica de substituição não populacional, pode ser encontrada em CHU et al. (1997).

3.3.4 A Meta-heurística GRASP

O algoritmo GRASP, cujo nome vem do inglês *Greedy Randomized Adaptive Search Procedure*, é uma das meta-heurísticas que juntamente com *tabu search* foram desenvolvidos usando apenas conceitos existentes no campo da pesquisa operacional. Deve-se lembrar que Glover, o inventor das meta-heurísticas de *tabu search* e busca dispersa, já era um pesquisador conceituado no desenvolvimento de técnicas de otimização, especialmente no desenvolvimento de técnicas clássicas de otimização. Os inventores do GRASP, Feo e Resende, também já eram pesquisadores conceituados em temas de otimização no campo da pesquisa operacional. Assim, eles tinham pleno domínio da teoria e aplicação das técnicas heurísticas de otimização, assim como das técnicas exatas de otimização (as técnicas clássicas de otimização). Particularmente, Feo e Resende conheciam o algoritmo heurístico construtivo e a heurística SDH. Assim, uma análise detalhada do GRASP permite verificar de que esse algoritmo é apenas uma junção e uma generalização do AHC de tipo guloso e da heurística SDH. Na análise das heurísticas tradicionais, que apareceram muito antes que as meta-heurísticas, verifica-se que o AHC de tipo guloso e a heurística SDH eram os mais importantes pelo desempenho que apresentam e porque essas heurísticas podiam ser usadas para inventar meta-heurísticas através de um processo de junção e de generalização. Portanto, o algoritmo GRASP é apresentado e analisado usando os tópicos conceituais já desenvolvidos para o AHC de tipo guloso e da heurística SDH (ROMERO; LAVORATO, 2012).

O algoritmo GRASP, para um problema genérico, assume a seguinte forma:

1. Passo preliminar: Montar os dados do problema. Escolher uma forma de representação de uma proposta de solução denominada de p . Identificar uma forma de avaliar a qualidade da função objetivo ou equivalente e denominada $f(p)$. Escolher uma estratégia heurística

construtiva (um algoritmo heurístico de tipo guloso a ser usado na fase construtiva) e uma estratégia de busca local para ser usada na fase de busca local (implica escolher uma heurística tipo SDH ou mesmo outra meta-heurística, juntamente com as estruturas de vizinhança e os correspondentes parâmetros).

2. Implementar a fase de pré-processamento.
3. Realizar a fase de busca construtiva.
4. Realizar a fase de pós-processamento de busca local. Atualizar a incumbente caso seja possível.
5. Se o critério de parada não for satisfeito, voltar ao passo 2. Caso contrário, pare. A resposta do algoritmo é a incumbente armazenada.

O pré-processamento tenta identificar determinadas subestruturas, isto é, atributos ou conjunto de atributos, que permitem iniciar o processo de busca construtiva de forma mais eficiente ou diminuir o espaço de busca do problema. A ideia central na fase de pré-processamento é diminuir o espaço de busca incorporando componentes do problema na solução em construção porque existe a certeza ou quase certeza de que esses componentes fazem parte da solução e descartando outros componentes porque existe certeza ou quase certeza de que esses componentes não devem fazer parte da solução ótima ou quase ótima. Essa estratégia é muito comum e muito usada na otimização clássica, especialmente nas técnicas de solução de problemas de programação (linear) inteira mista. Nesse caso existem várias estratégias de pré-processamento que permitem fixar variáveis inteiras ou binárias nos valores ótimos antes de iniciar o processo de otimização (por exemplo combinando restrições em programação binária). Também existem estratégias de geração de restrições que eliminam uma parcela da região factível, porque existe a certeza de que a solução ótima não se encontra nessa parcela da região factível descartada (por exemplo, as restrições substitutas). Adicionalmente, essa estratégia de fixação de variáveis ou redução do espaço de busca, não somente faz parte de uma fase de pré-processamento. Essa estratégia em alguns casos faz parte da própria estratégia central de uma técnica de otimização. Assim, por exemplo, a estratégia central do algoritmo *branch and bound* consiste em relaxar a região factível do problema original (ao relaxar a integralidade das variáveis inteiras) e depois a ideia central é gerar novas restrições para separar de forma disjuntiva a região factível do problema relaxado e diminuir sistematicamente o tamanho dessas regiões factíveis reduzidas até encontrar a solução ótima do problema original. Esse tipo de ideia é usada na formulação da estrutura VNS desenvolvido neste trabalho para resolver o problema de programação ótima de grade horária.

A fase construtiva do GRASP é apenas um algoritmo heurístico construtivo (AHC) de tipo guloso generalizado. Portanto, na fase construtiva do GRASP, a ideia central é gerar uma solução de boa qualidade tentando contornar as limitações do AHC de tipo guloso. Adicionalmente, a proposta permite gerar um número elevado de soluções de qualidade e geralmente diferentes, quando o algoritmo é processado repetidas vezes. Para entender a utilidade da fase construtiva do GRASP, deve-se analisar de forma resumida as limitações do AHC de tipo guloso. Um AHC de tipo guloso gera uma solução, geralmente de boa qualidade, através de uma sequência de passos e, em cada passo, adiciona-se uma componente da solução em construção, como mencionamos anteriormente. A componente que é incorporada na solução em construção é identificada usando

um indicador de sensibilidade. No AHC, a componente escolhida é a primeira classificada na ordem de mérito. A fase construtiva do GRASP simplesmente generaliza essa ideia central escolhendo não necessariamente a primeira classificada na ordem de mérito, mas escolhendo uma componente de um conjunto reduzido dos melhores classificados. Essa lista dos melhores classificados é chamada de lista RCL e deve ser de tamanho variável em cada iteração. Em resumo, a fase construtiva do GRASP tenta contornar as limitações de um AHC e também permite gerar um número elevado de propostas de solução diferentes. A lógica fundamental do GRASP na fase construtiva consiste em escolher a próxima componente da solução em construção dentre uma lista reduzida de candidatos, chamada de lista RCL. Um algoritmo heurístico construtivo escolheria o primeiro elemento dessa lista RCL. O número de elementos da lista RCL deve ser variável e deve levar em conta a qualidade dos componentes candidatos a adição. Assim, esse número de elementos depende da qualidade das componentes candidatas a serem incorporadas na solução em construção e do valor de um parâmetro α que deve ser escolhido.

A fase construtiva do algoritmo GRASP apresenta os seguintes passos:

1. Escolher a solução inicial que pode ser vazia, isto é, sem adição de componentes, que se transforma na solução em construção.
2. Para a solução em construção (com alguns elementos já adicionados) e usando um indicador de sensibilidade, elaborar uma lista RCL com os k componentes mais atrativos. O valor de k é determinado de forma adaptativa como é mostrado adiante.
3. Escolher um elemento (componente) dos k elementos existentes na lista RCL e atualizar a solução em construção corrente com a adição da componente escolhida.
4. Se a solução em construção corrente representa uma solução factível ou foi satisfeito o critério de parada (sem encontrar uma solução factível) terminar com a fase construtiva. Caso contrário voltar ao passo 1.

Os passos da fase construtiva do GRASP estão claramente apresentados sendo que precisamos apenas analisar a forma em que se encontra os elementos da lista reduzida RCL e a forma de escolha de um elemento dessa lista. Assim, inicialmente, mostra-se uma forma adequada para encontrar os elementos da lista RCL. Seja $h(x)$ uma medida de sensibilidade da função objetivo de um problema de minimização. Nesse contexto, fazem parte do conjunto RCL todos os candidatas cujos índices satisfazem a seguinte relação:

$$RCL = \{i \in X / h^{min} \leq h_i(x) \leq h^{min} + \alpha(h^{max} - h^{min})\}$$

em que X é o conjunto de índices das componentes que ainda podem ser adicionadas e α é um parâmetro fornecido pelo usuário com valores $0 \leq \alpha \leq 1$. h^{max} e h^{min} são os valores máximo e mínimo do medidor de sensibilidade dentre os elementos candidatos a adição na solução em construção. Deve-se observar que, em cada iteração, mudam os valores de h^{max} e h^{min} e, portanto, mesmo usando um valor fixo de α , o tamanho da lista RCL muda em cada iteração. Logicamente, um algoritmo heurístico construtivo (guloso) escolheria a componente relacionada

com $h_i(x) = h^{min}$. O parâmetro α representa um compromisso entre escolha aleatória e gulosa e, pode-se verificar que $\alpha = 1$ representa um processo totalmente aleatório e $\alpha = 0$ um processo guloso.

Uma vez montados os elementos da lista RCL, então o próximo passo é a escolha de um elemento dessa lista e que deve ser adicionada na solução em construção. Essa escolha pode ser realizada de duas formas: (1) aleatoriamente e, (2) usando uma função de distribuição de probabilidade como, por exemplo, a função de distribuição de probabilidade linear $b_i = 1/r_i$ em que r_i representa a posição que ocupa a componente i na lista de ordenada em RCL de acordo com a qualidade do indicador de sensibilidade (a melhor é a primeira da lista). Portanto, a probabilidade de escolha do elemento i é encontrada usando a relação:

$$p_i = \frac{b_i}{\sum_{j \in RCL} b_j}$$

Na formulação inicial do GRASP a fase de pós-processamento ou de busca local era realizado por uma heurística de do tipo SDH. A ideia central é tentar encontrar uma solução ótima local na vizinhança da solução encontrada na fase construtiva. Através do tempo o GRASP foi sofisticando a proposta de otimização na fase de busca local, após a fase construtiva. Assim, nas propostas iniciais a proposta era usar uma heurística de busca local como a heurística SDH ou como o processo de intensificação no algoritmo *tabu search*. Em geral, a fase de busca local pode ser um processo muito simples ou muito complexo. Assim, no caso mais simples pode ser implementado uma heurística tipo SDH. No caso mais sofisticado pode ser outra meta-heurística tais como *simulated annealing*, o algoritmo genético, *tabu search*, VNS, etc. Propostas intermediárias podem incorporar estratégias de otimização que fazem parte de meta-heurísticas sofisticadas como, por exemplo, a estratégia de *path relinking* que foi inventada como sendo parte do *tabu search*.

Em resumo, o algoritmo GRASP é uma integração e generalização sofisticada de dois algoritmos que já existiam no campo da pesquisa operacional, isto é, do algoritmo heurístico de tipo guloso e da heurística SDH. Em resumo, o algoritmo GRASP apresenta as seguintes características fundamentais:

- Uma fase de pré-processamento inspirada em conceitos existentes na pesquisa operacional clássica e cujo objetivo é incorporar ou descartar elementos que formam parte da solução ótima e, portanto, reduzindo o espaço de busca.
- Uma fase construtiva que na verdade é um algoritmo heurístico construtivo generalizado, onde se tenta contornar a característica gulosa do algoritmo heurístico construtivo e ter a possibilidade de gerar um conjunto elevado de soluções de qualidade usando uma mesma heurística construtiva.
- Uma fase de pós-processamento que na verdade é uma fase de melhoria da solução encontrada na fase construtiva que originalmente era implementado usando uma heurística simples de busca através de vizinhança como a heurística SDH. Entretanto, essa fase de busca local

pode ser implementada usando heurísticas ou meta-heurísticas sofisticadas como qualquer outra meta-heurística existente na literatura especializada.

Adicionalmente, como a fase construtiva e de busca local podem ser repetidos, então existe grande possibilidade de encontrar uma solução ótima global ou soluções quase-ótimas. Informação muito importante sobre a teoria e aplicação do GRASP pode ser encontrada em RESENDE (2008).

Existem muitas outras meta-heurísticas na literatura especializada. Entre as principais e que não foram abordadas neste Capítulo se encontram o algoritmo VNS (*Variable Neighborhood Search*), o algoritmo PSO (*Particle Swarm Optimization*), o algoritmo de busca dispersa (*Scatter Search*), o algoritmo de colônia de formigas, entre outros. Para uma revisão geral sobre as meta-heurísticas clássicas ver KOCHENBERGER (1994). Finalmente, deve-se observar que na última década apareceram novas meta-heurísticas na literatura especializada. O algoritmo VNS, usado neste trabalho, é abordado em detalhe no próximo capítulo.

Capítulo 4

A Meta-heurística VNS

A meta-heurística VNS (*Variable Neighborhood Search*) foi inventada por Hansen e Mladenovic. Informação muito importante sobre a teoria e aplicação do VNS pode ser encontrada em HANSEN (1986), HANSEN (2001) e HANSEN-MLADENOVIC (2001). Trata-se de uma meta-heurística inspirada apenas em conceitos típicos existentes em otimização matemática da mesma forma como acontece com *tabu search*, *scatter search* e GRASP. A estratégia central da meta-heurística é realizar a busca da solução ótima através do espaço de busca mudando a vizinhança sempre que for necessário. Dessa forma, o VNS também pode ser considerado como sendo uma generalização da heurística SDH. Deve-se observar que a heurística SDH usa apenas um tipo de vizinhança e o processo de otimização termina quando a melhor solução vizinha é pior que a proposta de solução corrente.

Uma generalização da heurística SDH pode ser passar a analisar uma vizinhança de outro tipo quando a melhor solução vizinha é pior que a solução corrente. Essa nova vizinhança pode ser definida de outra forma e geralmente deve ter uma cardinalidade maior. Assim, se for encontrada uma solução vizinha de melhor qualidade nesse novo tipo de vizinhança, então, deve-se realizar a transição e voltar a busca na vizinhança inicial. Uma generalização dessa proposta levou a inventar a meta-heurística VNS mais simples, chamada de VND (*Variable Neighborhood Descent*).

A maioria das meta-heurísticas aceitam a degradação da solução corrente (ou do conjunto de soluções correntes) como uma estratégia para sair de uma solução ótima local. O algoritmo VNS não aceita essa possibilidade. O algoritmo VNS muda a vizinhança como uma forma de sair de soluções ótimas locais. Nesse processo, a solução corrente também é a incumbente o que não acontece nas outras meta-heurísticas. Assim, pode-se afirmar que o algoritmo VNS realiza um conjunto de transições no espaço de busca de um problema e em cada passo a transição é realizada para a nova incumbente. Se o processo encontra um ótimo local, então o algoritmo VNS muda de vizinhança para sair desse ótimo local e passar para a nova incumbente. Como uma consequência dessa estratégia, se o algoritmo VNS encontra o ótimo global, então a busca fica estagnada nesse ponto de ótimo global sem possibilidade de sair desse ponto. Esse tipo de comportamento não acontece com as outras meta-heurísticas.

A estratégia do algoritmo VNS está inspirada em três observações importantes (MLADENOVIC, 1995):

- **Fato 1:** Um ótimo local para uma estrutura de vizinhança não necessariamente é ótimo local para outra estrutura de vizinhança.
- **Fato 2:** Um mínimo global é um mínimo local em relação a todos os tipos de estrutura de vizinhança.
- **Fato 3:** Para muitos problemas o mínimo local em relação a uma estrutura de vizinhança apresenta muitas particularidades com os mínimos locais de outras estruturas de vizinhança.

Os três fatos mostrados anteriormente são muito importantes para a sistematização da estratégia do algoritmo VNS. Os dois primeiros fatos são muito triviais. Assim, o fato 1 apenas menciona que se para a solução corrente for definida duas estruturas de vizinhanças diferentes, então a melhor solução vizinha encontrada com uma estrutura de vizinhança não necessariamente é a melhor solução vizinha encontrada usando a outra estrutura de vizinhança. O fato 2 menciona que se for definida uma estrutura de vizinhança para a solução corrente em que a melhor solução vizinha é o mínimo global, então esse mínimo para essa vizinhança deve ser melhor que a melhor solução vizinha de qualquer outra estrutura de vizinhança. Adicionalmente, o fato 3 é particularmente importante na formulação de um algoritmo VNS. A observação de caráter empírica implica que uma solução ótima local fornece informação importante em relação ao ótimo global especialmente se a solução ótima local for de excelente qualidade. Existe também a observação empírica de que as soluções ótimas locais geralmente estão concentradas em regiões específicas do espaço de busca. Se as soluções ótimas locais estivessem uniformemente distribuídas no espaço de busca, então todas as meta-heurísticas se tornariam ineficientes. Portanto, se for encontrado um ótimo local da região em que se encontra o ótimo global, então uma meta-heurística tipo VNS tem grande chance de encontrar esse ótimo global. Por outro lado, se o ótimo global se encontra em outra região distante, então a única possibilidade de encontrar o ótimo global é implementar um processo de diversificação. Por esse motivo um equilíbrio entre intensificação e diversificação no processo de busca pode ser importante em uma meta-heurística. Deve-se observar que esses fatos conceituais já foram mencionados anteriormente por Glover na formulação do algoritmo *tabu search*.

Existe um outro aspecto importante que deve formar parte da lógica de implementação de um algoritmo VNS que está relacionado com a qualidade de um ótimo local. Um ótimo local de função objetivo de melhor qualidade não necessariamente pode ser mais adequado para tentar encontrar um ótimo global. Supor que existem duas soluções ótimas locais x_a e x_b em que $f(x_a) < f(x_b)$ para o problema de minimização. Na análise tradicional podemos concluir que x_a é um ótimo local de melhor qualidade que x_b . Entretanto, se essas soluções devem ser usadas para iniciar (ou reiniciar) o processo de busca então podemos afirmar que aquela solução com características internas mais próximas da solução ótima global é a mais adequada para iniciar (ou reiniciar) a busca e, portanto, não necessariamente a solução x_a deve ser escolhida. Assim, por exemplo, para o problema de otimização de grade horária, a proposta de solução que tiver o maior número de elementos na mesma posição da grade ótima é a mais adequada para iniciar (ou reiniciar) a busca. Obviamente essa observação tem apenas utilidade conceitual

já que, em condições normais, não conhecemos a configuração da solução ótima. Entretanto, existem problemas onde a solução ótima é conhecida e existem vários algoritmos heurísticos para encontrar soluções ótimas locais para esse problema. Nesse contexto, podemos usar a observação anterior para identificar o algoritmo heurístico que produz soluções ótimas locais de melhor qualidade para iniciar a busca usando o algoritmo VNS. Assim, pode-se identificar o melhor algoritmo heurístico construtivo para ser incorporado na estrutura de solução de um algoritmo VNS.

Existem várias formas de implementar o algoritmo VNS e, portanto, podem ser implementados uma família de algoritmos VNS. Assim, apresenta-se essas versões do algoritmo VNS em ordem de complexidade. Denota-se por N_k , ($k = 1, \dots, kmax$), um conjunto finito de estruturas de vizinhança pré-selecionadas e seja $N_k(x)$ o conjunto de soluções ou vizinhos na k -ésima vizinhança de x . Uma solução ótima x_{opt} (ou mínimo global) é uma solução onde o mínimo é alcançado. Assim, x' é um mínimo local com relação a N_k , se não existir solução, na vizinhança especificada, tal que $f(x) < f(x')$. Portanto, a ideia é definir um conjunto de estruturas de vizinhança que podem ser usadas de forma determinística, de forma aleatória ou determinística e aleatória. Essas formas de usar a estrutura de vizinhança produzem algoritmos VNS de desempenhos diferentes.

4.1 O Algoritmo VND

Existem várias propostas de algoritmos VNS que podem ser usadas de forma independentes ou integradas em estruturas VNS mais complexas. A forma mais simples de um algoritmo tipo VNS é o algoritmo *Variable Neighborhood Descent (VND)*. O algoritmo VND está inspirado no Fato 1, mencionado anteriormente, isto é, um mínimo local para um tipo de vizinhança não necessariamente é o mínimo local para outro tipo de vizinhança. Assim, o ótimo local x' de x na vizinhança $N_1(x)$ não necessariamente é igual ao ótimo local x'' de x para a vizinhança $N_2(x)$. O algoritmo VND assume a seguinte estrutura:

1. **Passo preliminar:** Montar os dados do problema. Selecione um conjunto de estruturas de vizinhança N_s , $s = 1, \dots, s_{max}$, que devem ser usados na estrutura VND. Escolha uma forma de representar uma proposta de solução denominada p . Identificar uma forma de avaliar a qualidade da função objetivo ou equivalente e denominada $f(p)$ e, uma forma de identificar propostas de soluções ineficazes. Encontre uma solução inicial p_o que se transforma na solução corrente p_c .
2. Faça $s = 1$.
3. Encontre a melhor solução vizinha p^{best} da solução corrente p_c com $p_c \in N_s(p_c)$.
4. Se p^{best} é melhor que p_c , então faça $p_c = p^{best}$, $s = 1$ e volte ao passo 3. Em caso contrário, passar ao passo 5.
5. Faça $s = s + 1$. Se $s \leq s_{max}$ volte ao passo 3. Em caso contrário, termine o processo de otimização.

O algoritmo VND representa a forma mais simples de formular um algoritmo VNS que

prioriza a lógica de intensificação. Para determinadas aplicações pode ser muito eficiente. Adicionalmente, esse tipo de algoritmo básico pode ser integrado em uma estrutura mais complexa de algoritmo VNS. Quando o algoritmo VND for usado de forma independente deve-se priorizar a busca de soluções de excelente qualidade. Por outro lado, se o algoritmo VND é usado em estruturas mais complexas pode ser mais importante encontrar uma boa solução mais rapidamente. Deve-se observar que o algoritmo VND representa uma generalização da heurística SDH.

Em relação ao algoritmo VND, deve-se fazer as seguintes observações:

- O tamanho da vizinhança é um assunto crítico. Se o tamanho das vizinhanças for muito grande, então o algoritmo VND pode se tornar muito lento. Assim, se o algoritmo VND for usado de forma independente, então podem ser aceitáveis vizinhanças de cardinalidade elevada (um número elevado de vizinhos). Por outro lado, se o algoritmo VND for usado em uma estrutura VNS mais complexa (ver, por exemplo, o algoritmo GVNS), então as vizinhanças devem ter cardinalidade não muito grande. Em qualquer contexto, pode ser recomendável usar técnicas de redução de vizinhança da mesma forma que no algoritmo *tabu search*.
- Em relação com o item anterior, pode-se modificar o algoritmo VND para que não seja muito lento. Observa-se que no passo 3, todos os vizinhos são avaliados para identificar o melhor vizinho. Assim, apenas o melhor vizinho é comparado com a proposta de solução corrente p_c (essa estratégia é a mesma da heurística SDH). Esse passo pode ser mudado de forma que os vizinhos sejam ordenados e assim que for identificado um vizinho melhor que a proposta de solução corrente, então, deve-se realizar a transição. Nesse contexto os tamanhos das vizinhanças não são críticos.
- Na lógica do algoritmo VND está implícito que as vizinhanças mudam de tamanho de forma crescente, isto é, a vizinhança N_1 tem menos elementos que a vizinhança N_2 e assim sucessivamente. Portanto, a estrutura do algoritmo VND prioriza a vizinhança de menor tamanho, isto é, sempre que for encontrada uma nova incumbente e após executada a transição, o algoritmo VND retorna (ou permanece) para a vizinhança N_1 . Essa proposta prioriza claramente a lógica da intensificação.
- As estruturas de vizinhança podem não ser suficientes para encontrar o ótimo global. Observa-se que as estruturas de vizinhança crescem em complexidade (cada nova estrutura tem um número muito maior de elementos ou vizinhos). Assim, se as estruturas de vizinhança não se encontram adequadamente projetadas, então pode não ser possível encontrar as soluções ótimas de problemas complexos. Portanto, definir de forma adequada as estruturas de vizinhança é crucial em algoritmos tipo VNS.
- Assim como na heurística SDH e as meta-heurísticas tais como *tabu search* e *simulated annealing*, o algoritmo VND (e todos os algoritmos tipo VNS) precisa de uma proposta de solução inicial p_o para iniciar o processo de otimização. No caso do VND, recomenda-se que essa proposta de solução inicial seja de boa qualidade (ótimo local). Portanto, essa proposta de solução inicial pode ser encontrada usando um AHC de bom desempenho.
- O algoritmo VND termina quando não existe proposta de solução vizinha de melhor qualidade que a solução corrente no último nível de vizinhança $N_{s_{max}}$. Portanto, o algoritmo

VND não precisa da especificação de um critério de parada, mas pode ser especificada uma estratégia de parada alternativa.

4.2 O Algoritmo RVNS

O segundo tipo de algoritmo VNS é o chamado *Reduced Variable Neighborhood Search* (RVNS). Este tipo de algoritmo está inspirado em dois aspectos fundamentais no processo de busca relacionados com a intensificação e a diversificação. Por um lado o Fato 3 afirma que na região de um ótimo local normalmente existem outras soluções ótimas locais que podem ser encontradas a partir de um ótimo local inicial e, portanto, deve-se montar uma estratégia de intensificação para tentar encontrar esses ótimos locais. Por outro lado, sair de um ótimo local de qualidade para encontrar um ótimo local de uma região mais distante exige uma estratégia que implique em uma mudança mais radical na caracterização da vizinhança, especialmente em problemas onde um ótimo local tem uma composição complexa. Assim, uma busca que contemple ambos os aspectos (intensificação e diversificação) pode permitir encontrar ótimos locais de uma mesma região e pode permitir ao processo de busca sair para ótimos locais de regiões mais distantes do ponto de ótimo local corrente. Esse tipo de estratégia, isto é, uma composição entre intensificação e diversificação foi incorporado no algoritmo RVNS. O algoritmo RVNS assume a seguinte estrutura:

1. **Passo preliminar:** Montar os dados do problema. Selecione um conjunto de estruturas de vizinhança N_s , $s = 1, \dots, s_{max}$, que devem ser usados na estrutura VND. Escolha uma forma de representar uma proposta de solução denominada p . Identificar uma forma de avaliar a qualidade da função objetivo ou equivalente e denominada $f(p)$ e, uma forma de identificar propostas de solução inactiváveis. Encontre uma solução inicial p_o que se transforma na solução corrente p_c . Escolher um critério de parada.
2. Faça $s = 1$.
3. Encontre de forma aleatória uma solução vizinha p^v da solução corrente p_c com $p_c \in N_s(p_c)$.
4. Se p^v é melhor que p_c , então faça $p_c = p^v$, $s = 1$ e volte ao passo 3. Em caso contrário, passar ao passo 5.
5. Faça $s = s + 1$. Se $s \leq s_{max}$ volte ao passo 3. Em caso contrário, passar ao passo 6.
6. Se o critério de parada for satisfeito, termine o processo de otimização. Em caso contrário ir ao passo 2.

Deve-se observar que o algoritmo RVNS produz uma escolha de vizinhos mais dinâmica escolhendo vizinhos de todas as estruturas de vizinhança (diversificação) e priorizando a primeira estrutura de vizinhança (intensificação) sempre que for realizada a transição da proposta de solução corrente para uma proposta de solução vizinha de melhor qualidade. Entretanto, uma componente importante da estrutura RVNS é a capacidade de encontrar novas regiões promissoras a partir de um ótimo local. Adicionalmente, deve-se observar que o tamanho de cada

estrutura de vizinhança não é um problema crítico no algoritmo RVNS como pode acontecer no algoritmo VND. O algoritmo RVNS também pode ser usado de forma independente ou pode ser integrado em uma estrutura mais complexa de algoritmo VNS.

Em relação ao algoritmo RVNS, deve-se fazer as seguintes observações:

- No passo 3, é escolhida de forma aleatória uma solução vizinha da solução corrente para uma certa estrutura de vizinhança. Deve-se observar que a solução corrente é um ótimo local (incumbente). Nesse contexto, a estratégia RVNS procura, de forma aleatória, uma solução vizinha de melhor qualidade usando as estruturas de vizinhança simples (intensificação) e também uma solução vizinha de melhor qualidade usando as estruturas de vizinhança mais complexas (diversificação).
- Outro aspecto importante e crucial é quanto se deve insistir usando uma estrutura de vizinhança na busca de um ótimo local. Por um lado, existe o Fato 3 que estipula que em muitos problemas um ótimo local geralmente se encontra muito próximo de outras soluções ótimas locais e localizado em pequenas regiões do espaço de busca. Portanto, quando for encontrado um ótimo local, então essa solução contém informação implícita das outras soluções ótimas locais e, provavelmente, da própria solução ótima global. Assim, é natural explorar de forma prioritária nas proximidades de um ótimo local. Entretanto, se essa região de ótimo local se encontra muito distante de outras regiões de ótimo local, então a estratégia de busca local pode não ser suficiente para sair dessa região de ótimo local. Portanto, pode ser necessário adicionar uma estratégia mais dinâmica no processo de busca. Essa possibilidade é introduzida no algoritmo RVNS.
- No algoritmo RVNS, escolhe-se um conjunto de estruturas de vizinhança. Geralmente as estruturas de vizinhança estão aninhadas, isto é, todos os elementos da estrutura de vizinhança N_1 também fazem parte dos elementos da estrutura de vizinhança N_2 . Em alguns problemas não existe esse tipo de relação. Assim, sempre que for realizada uma transição da solução corrente para a nova solução corrente, então o processo volta para o primeiro nível de vizinhança. Em caso de insucesso, então o processo passa para o nível superior de vizinhança. Nesse processo, caso seja atingido o último nível de vizinhança, deve-se retornar para o primeiro nível de vizinhança em qualquer contexto (com transição e sem transição). Por esse motivo, o algoritmo RVNS precisa de um critério de parada.
- O tamanho das estruturas de vizinhança não é um problema crítico no algoritmo RVNS já que em cada nível de vizinhança é escolhida uma solução vizinha de forma aleatória para tentar realizar a transição. Por esse motivo o algoritmo RVNS tem uma componente estocástica e em cada corrida podem ser encontradas soluções ótimas locais diferentes. Deve-se observar que o algoritmo VND é integralmente determinístico, isto, uma vez escolhidas a proposta de solução inicial e as estruturas de vizinhança, o algoritmo encontra a mesma solução final em qualquer corrida do algoritmo.
- Assim como o algoritmo VND o algoritmo RVNS precisa de uma proposta de solução inicial p_o para iniciar o processo de otimização. Assim, recomenda-se que essa proposta de solução inicial seja de boa qualidade. Portanto, essa proposta de solução inicial pode ser encontrada usando um AHC de bom desempenho.
- O algoritmo RVNS precisa definir um critério de parada claramente especificado.

4.3 O Algoritmo BVNS

Algoritmos VNS mais eficientes podem ser formulados integrando as características do algoritmo VND, que permite encontrar ótimos locais de qualidade, e do algoritmo RVNS que permite encontrar novas regiões promissoras a partir de um ótimo local. Assim, juntando as características dos dois algoritmos já apresentados podem ser formulados dois tipos de algoritmos VNS que geralmente apresentam excelente desempenho. Esses algoritmos são chamados de *Basic Variable Neighborhood Search* (BVNS) e *General Variable Neighborhood Search* (GVNS).

O algoritmo BVNS combina a busca local com mudanças sistemáticas da vizinhança em torno de uma solução ótima local. A estrutura do algoritmo BVNS assume a seguinte forma:

1. **Passo preliminar:** Montar os dados do problema. Selecione um conjunto de estruturas de vizinhança N_s , $s = 1, \dots, s_{max}$, que devem ser usados na estrutura BVNS. Escolha uma forma de representar uma proposta de solução denominada p . Identificar uma forma de avaliar a qualidade da função objetivo ou equivalente e denominada $f(p)$ e, uma forma de identificar propostas de solução inactivíveis. Encontre uma solução inicial p_o que se transforma na solução corrente p_c . Escolher um critério de parada.
2. Faça $s = 1$.
3. Encontre de forma aleatória uma proposta de solução vizinha p^v da proposta de solução corrente p_c no nível de vizinhança N_s .
4. Implementar uma fase de busca local a partir da proposta de solução p^v escolhida aleatoriamente. Seja p^{nov} o ótimo local encontrado no processo de busca local a partir de p^v .
5. Se p^{nov} é melhor que p_c , então faça $p_c = p^{nov}$, $s = 1$ e volte ao passo 3. Em caso contrário, passar ao passo 6.
6. Faça $s = s + 1$. Se $s \leq s_{max}$ volte ao passo 3. Em caso contrário, passar ao passo 7.
7. Se o critério de parada for satisfeito, termine o processo de otimização. Em caso contrário ir ao passo 2.

A lógica de trabalho do algoritmo BVNS é muito interessante. Inicialmente, deve-se escolher as s estruturas de vizinhança. O processo de otimização é iniciado de uma proposta de solução p_c e na vizinhança $N_1(p_c)$. Na sequência, escolhe-se de forma aleatória um vizinho p^v de p_c em $N_1(p_c)$. A partir de p^v é iniciado um processo de busca local para encontrar um ótimo local p^{nov} . Nesse contexto podem acontecer 3 casos: (1) se $p^{nov} = p_c$ significa que p_c já era o ótimo local e, portanto, deve-se mudar para outro nível de vizinhança ($N_2(p_c)$ neste caso); (2) se p^{nov} é de pior qualidade que p_c , então foi encontrado um ótimo local de pior qualidade que a incumbente p_c e também devemos mudar de vizinhança; e (3) se p^{nov} é de melhor qualidade que p_c , então significa que foi encontrada uma solução melhor que a incumbente e, portanto, deve-se atualizar

a incumbente e reiniciar a busca a partir da nova incumbente e permanecendo na vizinhança $N_1(p_c)$). Assim, em qualquer iteração do processo e sempre que o processo de busca local encontra uma nova incumbente, então o processo volta para a vizinhança $N_1(p_c)$ e sempre que o processo de busca local encontra uma solução de igual ou de pior qualidade que a incumbente, então se deve passar para uma vizinhança mais complexa. Essa estratégia e a escolha aleatória do vizinho da incumbente p_c permite encontrar ótimos locais distantes da incumbente corrente.

Em relação ao algoritmo BVNS, deve-se fazer as seguintes observações:

- No passo 3, é escolhida de forma aleatória uma solução vizinha da solução corrente para uma certa estrutura de vizinhança. Essa proposta de solução vizinha escolhida de forma aleatória é usada como solução inicial de um processo de otimização de busca local. Assim, essa escolha aleatória fornece um comportamento estocástico ao algoritmo BVNS.
- No passo 4 é implementado um processo de otimização chamado de busca local. Em princípio, pode-se implementar qualquer estratégia de busca local. Entretanto, fica evidente que na formulação inicial do algoritmo BVNS, essa estratégia de busca local seria a heurística SDH.
- O tamanho das estruturas de vizinhança não é um problema crítico no algoritmo BVNS já que em cada nível de vizinhança é escolhida uma solução vizinha de forma aleatória para iniciar o processo de otimização de busca local. Assim, o tempo de processamento do algoritmo BVNS depende mais fortemente da estratégia implementada na fase de busca local no passo 4 (da forma de definir a vizinhança se for usada a heurística SDH).
- Assim como o algoritmo VND e RVNS, o algoritmo BVNS precisa de uma proposta de solução inicial p_o para iniciar o processo de otimização. Assim, recomenda-se que essa proposta de solução inicial seja de boa qualidade. Portanto, essa proposta de solução inicial pode ser encontrada usando um AHC de bom desempenho.
- O algoritmo BVNS precisa definir um critério de parada claramente especificado.

4.4 O Algoritmo GVNS

O algoritmo *General Variable Neighborhood Search* (GVNS) representa a estrutura algorítmica mais complexa dentre os algoritmos da família VNS. O GVNS representa uma generalização do algoritmo BVNS onde aparece claramente integradas as estruturas VND e RVNS. Obviamente, a implementação computacional de um algoritmo GVNS é muito mais demorada e, deve-se projetar de forma adequada as diferentes estruturas de vizinhança para encontrar um algoritmo GVNS que não extrapole o esforço computacional.

A estrutura do algoritmo GVNS assume a seguinte forma:

1. **Passo preliminar:** Montar os dados do problema. Selecione um conjunto de estruturas de vizinhança N_s , $s = 1, \dots, s_{max}$, que deve ser usado na estrutura geral GVNS. Selecione um

conjunto de estruturas de vizinhança $N_r, r = 1, \dots, r_{max}$ que deve ser usado na estrutura do algoritmo RVNS. Escolha um critério de parada para o algoritmo RVNS. Selecione um conjunto de estruturas de vizinhança $N_q, q = 1, \dots, q_{max}$ que deve ser usado na estrutura do algoritmo VND. Escolha uma forma de representar uma proposta de solução denominada p . Identificar uma forma de avaliar a qualidade da função objetivo ou equivalente e denominada $f(p)$ e, uma forma de identificar propostas de solução infactíveis. Encontre uma solução inicial p_o que se transforma na solução corrente p_c . Escolher um critério de parada.

2. Faça $r = 1$.
3. Gere uma proposta de solução p^v de forma aleatória a partir da solução corrente p_c no nível de vizinhança N_r .
4. Se p^v é melhor que p_c , então faça $p_c = p^v, r = 1$ e volte ao passo 3. Em caso contrário, passar ao passo 5.
5. Faça $r = r + 1$. Se $r \leq r_{max}$ volte ao passo 3. Em caso contrário, passar ao passo 6.
6. Se o critério de parada não for satisfeito, então voltar ao passo 2. Em caso contrário ir ao passo 7 (foi encontrada a melhor solução obtida pelo algoritmo RVNS e que deve ser usada como solução inicial para a estrutura GVNS)
7. Faça $s = 1$.
8. Encontre de forma aleatória uma proposta de solução vizinha p^v da proposta de solução corrente p_c no nível de vizinhança N_s .
9. Faça $q = 1$.
10. Encontre a melhor solução vizinha p^{best} da solução p^v com $p^v \in N_q(p^v)$.
11. Se p^{best} é melhor que p^v , então faça $p^v = p^{best}, q = 1$ e volte ao passo 10. Em caso contrário ir ao passo 12.
12. Faça $q = q + 1$. Se $q \leq q_{max}$, voltar ao passo 10. Em caso contrário ir ao passo 13 (foi encontrado um ótimo local pelo algoritmo VND).
13. Se p^v é melhor que p_c , então $p_c = p^v, s = 1$ e volte para o passo 8. Em caso contrário ir ao passo 14.
14. Faça $s = s + 1$. Se $s \leq s_{max}$ voltar ao passo 8. Em caso contrário ir ao passo 15.
15. Se o critério de parada for satisfeito, então pare o processo de otimização. Em caso contrário voltar ao passo 7.

Todas as observações realizadas para o algoritmo BVNS permanecem válidas no algoritmo GVNS. Entretanto, existem duas mudanças significativas. A primeira mudança está na fase de melhoria local da proposta de solução inicial realizada pelo algoritmo RVNS e a segunda mudança significativa é que a busca local, que estava presente no algoritmo BVNS, é realizada por um algoritmo VND.

Em relação ao algoritmo GVNS, deve-se fazer as seguintes observações:

- Antes de processar a parcela principal do algoritmo VNS é realizado um processo de melhoria da proposta de solução inicial fornecida no passo preliminar ao algoritmo VNS. Assim, a proposta de solução inicial p_o é melhorada usando um algoritmo RVNS, realizado nos passos 2 a 6 do algoritmo GVNS. Portanto, na estrutura GVNS, a solução inicial p_o não precisa ser necessariamente de excelente qualidade já que essa proposta de solução inicial é melhorada por um algoritmo RVNS. Adicionalmente, o algoritmo RVNS deve ser rápido já que a única finalidade desse algoritmo é melhorar a proposta de solução inicial e, portanto, deve ter estruturas de vizinhança não muito complexas.
- Nos passos 2 a 15 aparece a estrutura principal do algoritmo GVNS. Dentro dessa estrutura, nos passos 9 a 12 aparece a fase de melhoria local realizada por um algoritmo VND.
- O algoritmo VND (passos 9 a 12) também deve ser suficientemente rápido, isto é, não pode ter estruturas de vizinhança muito complexas para não onerar o tempo de processamento. Em resumo, as estruturas de vizinhanças devem ser calibradas de forma adequada. Dessa forma a estrutura de vizinhança mais complexa deve corresponder ao algoritmo VNS geral ($N_s, s = 1, \dots, s_{max}$) e a estrutura de vizinhança mais simples deve corresponder ao algoritmo VND ($N_q, q = 1, \dots, q_{max}$). Dessa forma, a estrutura de vizinhança de complexidade intermediária corresponde ao algoritmo RVNS ($N_r, r = 1, \dots, r_{max}$).
- Assim como o algoritmo VND, RVNS e BVNS, o algoritmo GVNS precisa de uma proposta de solução inicial p_o para iniciar o processo de otimização. Assim, recomenda-se que essa proposta de solução inicial seja de boa qualidade. Portanto, essa proposta de solução inicial pode ser encontrada usando um AHC de bom desempenho, mas neste caso essa escolha pode ser mais simples já que essa solução inicial é melhorada usando um algoritmo RVNS.
- O algoritmo GVNS precisa definir um critério de parada claramente especificado.

No próximo capítulo é apresentada a estrutura VNS para resolver o problema de otimização de grade horária.

Capítulo 5

Meta-heurística VNS Aplicado ao Problema de Otimização de Grade Horária

5.1 Introdução

Neste Capítulo, apresenta-se todos os detalhes relacionados com a implementação de uma meta-heurística para o problema de programação ótima de grade horária. Assim, inicialmente é discutido o assunto relacionado com a forma mais adequada de representação de uma proposta de solução. Após resolver esse assunto, deve-se mostrar a forma de calcular o valor da função objetivo e, portanto, verificar a qualidade de uma proposta de solução e, finalmente, a forma de verificar a factibilidade de uma proposta de solução. Esses tópicos inicialmente abordados podem ser de utilidade na implementação de qualquer meta-heurística.

Na segunda parte do Capítulo, apresenta-se a estrutura idealizada para o algoritmo VNS para resolver de forma eficiente o problema de programação ótima de grade horária. Nessa parcela da informação é de especial importância a forma em que é idealizada a estrutura de vizinhança e a forma em que se identifica a melhor solução vizinha de uma proposta de solução. Nessa implementação é usada o modelo matemático apresentado no Capítulo 2, após fixar uma parcela das variáveis de decisão e resolvendo o problema reduzido restante usando o *solver* CPLEX. Assim, a forma de idealizar a vizinhança e a forma de identificação da melhor solução vizinha é muito diferente das implementações tradicionais usadas em meta-heurísticas para identificar a melhor solução vizinha da solução corrente.

Adicionalmente são apresentados outros detalhes de implementação computacional e a justificativa de não usar o modelo matemático apresentado no Capítulo 2 para resolver de forma

integral o problema de programação ótima de grade horária.

5.2 Forma de Representação de uma Proposta de Solução

A forma de representar uma proposta de solução é o primeiro assunto e o mais importante a ser solucionado quando se pretende resolver um problema complexo usando meta-heurísticas. Essa proposta deve ser eficiente, de forma que permita encontrar facilmente o valor da função objetivo ou equivalente, verificar infactibilidades e implementar as estratégias de vizinhança ou operadores equivalentes.

Neste trabalho foram idealizadas duas formas de representar uma proposta de solução. A primeira proposta é apresentada na Figura 1 e a segunda proposta é apresentada na Tabela 3. Ambas as propostas armazenam a mesma informação, isto é, uma proposta de solução que permite identificar a qualidade e a factibilidade da proposta e sua implementação.

A primeira forma de representar uma proposta de solução, mostrada na Figura 1, está formada apenas por dois vetores, nho e nsa , de tamanho igual ao número de eventos. A informação do evento aparece de forma implícita. O vetor nho armazena a hora em que foi programado um evento e o vetor nsa armazena a sala em que foi programado um evento. Assim, da Figura 1, pode-se verificar que o evento No. 1 foi programada na hora 21 e na sala 3, o evento No. 2 foi programado na hora 43 e na sala 1 e assim sucessivamente.

Figura 1: Representação de uma proposta de solução.

$nho =$	21	$nsa =$	3	1
	43		1	2
	11		9	3
	34		4	4

	35		5	400
				Evento

Fonte: Elaborado pelo autor.

A segunda proposta de representar uma proposta de solução, mostrada na Tabela 3, está representada em uma estrutura matricial. Essa matriz tem um número de linhas igual ao número de horas da programação semanal e um número de colunas igual ao número de salas de aula. Pode-se verificar que a proposta de representação de uma proposta de solução realmente é uma grade horária, isto é, representa a proposta de solução que pode ser verificada e implementada facilmente. Aparentemente, a primeira proposta sugerida para representar uma proposta de solução

pode parecer mais interessante, mas a segunda proposta é mais compacta para o tamanho de problemas abordados neste trabalho.

Na maioria dos problemas abordados existem 10 salas de aula, 400 eventos e 45 horas de programação. Nesse contexto, a primeira proposta armazena 800 elementos (dois vetores de 400 elementos) e a segunda proposta armazena 450 elementos (uma matriz de 45 linhas e 10 colunas). Adicionalmente, a segunda proposta permite implementar de forma mais simples as estruturas de vizinhança. Por esse motivo, adotamos a segunda proposta, a forma matricial, como sendo a forma mais adequada para representar uma proposta de solução. Entretanto, dependendo da meta-heurística adotada e da forma em que se pretende calcular o valor da função objetivo e as infactibilidades, pode ser conveniente preservar e manter ambas as formas de representar uma proposta de solução. Em outras palavras, se existe disponível a informação de uma proposta de solução na forma apresentada na Tabela 3, então pode ser conveniente também montar a forma de representação mostrada na Figura 1 obtida a partir da informação na Tabela 3. Esse tipo de informação pode ajudar a determinar mais rapidamente algum tipo de infactibilidade como se mostra adiante.

Para mostrar a forma de calcular o número de restrições violadas de cada tipo, deve-se mostrar a estrutura de dados fundamentais e estruturas de dados adicionais montadas a partir das informações fundamentais. Neste trabalho, os dados existentes no site da Competição Internacional de Grade Horária foram adaptados para assumir a forma apresentada no Apêndice. Nessa estrutura, para cada evento se mostra o número de salas em que o evento pode ser programado e a indicação das salas específicas habilitadas. Também para cada evento se mostra o número de alunos matriculados e a indicação específica dos alunos matriculados no evento. Para manipulação da informação, esses dados são armazenados na seguinte estrutura de dados:

- A informação do número de salas de cada evento é armazenado no vetor a e as salas habilitadas é armazenada na matriz A (Figura 2).
- A informação do número de alunos matriculados em cada evento é armazenada no vetor b e os alunos matriculados em cada evento é armazenada na matriz B (Figura 3).

Para os dados da instância No. 1 mostrado no Apêndice a forma dos vetores a e b e das matrizes A e B são mostradas na Figura 2 e na Figura 3.

Neste trabalho, uma proposta de solução para o problema de programação ótima de grade horária é armazenada na matriz PG . Assim, a proposta de solução mostrada na Tabela 3 é chamada de matriz PG . Com essa informação de uma proposta de solução, deve-se calcular o número de restrições obrigatórias violadas com a utilização dos vetores a e b e das matrizes A e B . Para calcular o número de restrições secundárias violadas é necessário montar vetores e matrizes adicionais e também pode ser mais conveniente usar a forma de representação secundária, isto é, a proposta de representação de uma proposta de solução mostrada na Figura 1. Esses detalhes são apresentados adiante.

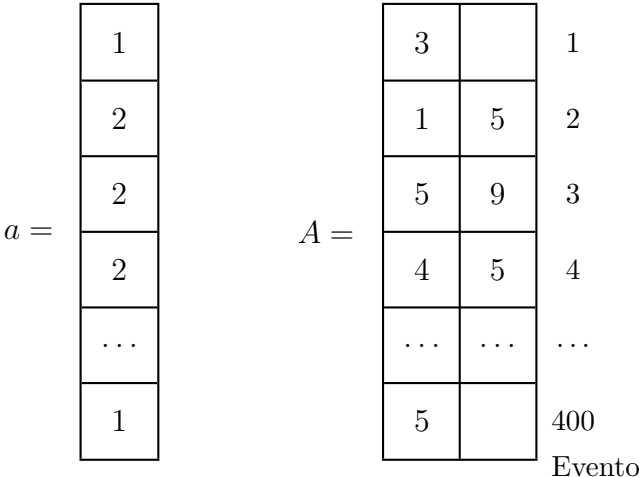
Tabela 3: Segunda proposta de representar uma proposta de solução.

Função objetivo: $v = 40$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	43	338	127	263	70	319	160	204	353	59
2	38	241	61	307	73	11	104	288	320	378
3	48	352	120	31	143	268	5	236	197	53
4	362	394	78	360	134	200	172	302	246	179
5	256	329	178	281	269	28	207	121	285	113
6	395	267	335	110	150	26	75	140	231	54
7	385	232	42	146	88	107	249	118	313	330
8	389	156	308	228	275	328	342	103	279	379
9	0	0	0	0	0	0	0	0	0	0
10	383	74	151	192	29	27	153	194	152	334
11	397	388	55	356	93	99	273	303	3	321
12	390	245	22	84	230	318	91	64	262	367
13	398	94	294	108	138	190	0	211	186	238
14	158	77	175	368	144	112	349	287	306	280
15	274	133	33	90	260	40	214	278	154	41
16	46	56	123	344	39	216	19	185	355	209
17	18	130	257	36	65	325	128	220	293	173
18	364	0	0	0	0	0	0	0	0	0
19	201	79	219	187	165	312	16	126	315	255
20	60	169	117	6	226	243	180	176	129	327
21	392	222	1	114	323	258	182	96	213	98
22	86	225	148	183	347	276	0	196	87	212
23	139	239	63	229	391	17	157	359	101	376
24	296	174	163	206	50	286	69	92	145	167
25	37	15	339	168	105	340	47	223	316	369
26	393	371	261	357	136	49	195	51	337	67
27	0	0	0	0	0	0	0	0	8	0
28	295	282	162	21	106	300	24	254	95	299
29	384	242	171	272	166	9	348	264	109	301
30	181	52	290	44	193	23	25	147	62	377
31	332	184	326	210	170	35	366	289	221	341
32	132	141	97	12	122	66	265	68	331	358
33	363	333	45	32	374	322	102	0	346	336
34	387	198	283	4	76	250	58	247	135	224
35	89	343	297	149	400	270	155	234	350	380
36	0	0	0	0	0	0	0	0	0	0
37	396	309	271	372	203	284	298	314	305	240
38	248	215	82	304	375	34	266	218	205	351
39	365	115	252	81	30	131	345	233	125	370
40	57	199	159	259	13	217	227	208	361	164
41	191	137	7	124	116	202	161	111	177	235
42	85	311	354	244	399	14	119	291	317	381
43	2	237	20	71	386	292	83	72	324	188
44	80	382	251	253	10	310	142	100	277	373
45	0	0	0	0	189	0	0	0	0	0

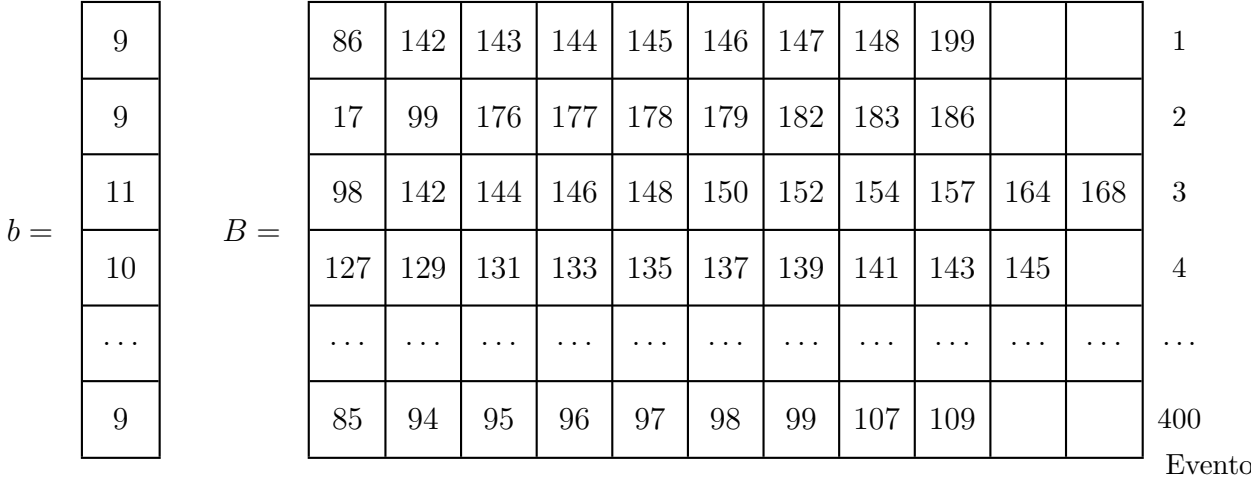
Fonte: Elaborado pelo autor.

Figura 2: Vetor a e matriz A do banco de dados.



Fonte: Elaborado pelo autor.

Figura 3: Vetor b e matriz B do banco de dados.



Fonte: Elaborado pelo autor.

5.3 Verificação das Restrições Obrigatórias e as Restrições Secundárias

Na implementação de meta-heurísticas, uma função objetivo equivalente do problema de otimização de grade horária pode minimizar as restrições obrigatórias e secundárias na seguinte forma:

$$\min v = \alpha NRO + \beta NRD$$

onde NRO é o número de restrições obrigatórias de uma proposta de solução, NRD é o número de restrições secundárias de uma proposta de solução e α e β são parâmetros que devem ser escolhidos. No problema de grade horária não é aceitável propostas de solução que violem as restrições obrigatórias e, por esse motivo, os valores adequados para os parâmetros são os seguintes: $\alpha = 1000$ e $\beta = 1$. Dessa forma, de maneira muito simples, pode-se verificar se uma proposta de solução é factível (não viola nenhuma das restrições obrigatórias) e a qualidade da proposta (o número de restrições secundárias violadas). Esta Seção mostra a forma de calcular o número de restrições violadas de cada tipo se existe disponível uma proposta de solução como a mostrada na Tabela 3 (ou na Figura 1). Como mencionado anteriormente, se existe disponível uma proposta de solução como a mostrada na Tabela 3, então pode ser conveniente montar a proposta de solução no formato da Figura 1.

A continuação, apresenta-se a forma de calcular o número de restrições violadas de casa tipo. Para mostrar a forma de calcular o número de restrições violadas de cada tipo, assume-se de que existe uma proposta de solução apresentada no formato da Tabela 3 (chamada de matriz PG) e a forma equivalente apresentada no formato da Figura 1 (os vetores nho e nsa). Adicionalmente, existe disponível os vetores a e b e as matrizes A e B . Nesse contexto o número de restrições violadas são encontradas usando a seguinte estratégia:

1. Número de restrições de tipo RO_1 violadas:

A restrição de tipo RO_1 exige que dois ou mais eventos não podem ser programados na mesma sala e no mesmo horário. O número de restrições do tipo RO_1 violadas se encontram de forma trivial a partir da matriz PG . Assim, se em cada uma das 450 posições da matriz PG existe apenas um elemento (um evento armazenado), então não existem restrições de tipo RO_1 violadas. Portanto, na forma idealizada para representar uma proposta de solução nunca aparecem restrições do tipo RO_1 violadas.

2. Número de restrições de tipo RO_2 violadas:

A restrição de tipo RO_2 exige que o evento seja programado em uma sala adequada. O número de restrições do tipo RO_2 violadas se encontram de forma rápida a partir da matriz PG . Assim, para cada evento programado em PG , pode-se verificar se está programado na sala adequada com a ajuda do vetor a e a matriz A .

3. Número de restrições de tipo RO_3 violadas:

A restrição de tipo RO_3 exige que um aluno não pode ser programado em mais de um evento no mesmo horário. O número de restrições do tipo RO_3 violadas pode ser encontrado a partir da informação em PG e no vetor b e a matriz B , mas o esforço de processamento pode ser elevado. Assim, para acelerar o processo de verificação de restrições de tipo RO_3 violados, deve-se montar o vetor c e a matriz C . Nesses arranjos se armazena o número de eventos programado para cada aluno e a especificação desses eventos. Na Figura 4 se apresenta a estrutura do vetor c e da matriz C para a instância No. 1 do problema de grade horária.

Figura 4: Vetor c e matriz C montado a partir do banco de dados.

$c =$	17	$C =$	24	35	50	101	161	175	231	243	248	257	268	273	279	283	324	364	374	1		
	17		15	23	24	35	67	101	128	143	161	175	198	243	251	273	279	324	374	2		
	17		24	31	35	81	101	161	175	224	225	231	243	253	257	273	279	324	374	3		
	18		23	24	31	81	91	101	105	128	161	175	243	246	273	279	283	324	374	382	4	
	
	18		10	13	23	24	35	94	101	128	161	175	224	243	268	273	302	324	353	374	200	
																						Aluno

Fonte: Elaborado pelo autor.

Da Figura 4 é possível verificar os eventos em que cada aluno está matriculado. Assim, por exemplo, o aluno No. 1 está matriculado em 17 eventos e na primeira linha da matriz C aparecem os eventos nos quais se encontra matriculado.

Para encontrar o número de restrições tipo RO_3 violadas correspondente a cada aluno, pode-se percorrer as linhas da matriz C e consultar no vetor nho se existe algum evento programado na mesma hora com outro evento. Deve-se observar que o número de restrições tipo RO_3 violadas pode também ser encontrado percorrendo a matriz PG e com a ajuda da informação armazenada no vetor b e na matriz B é possível encontrar o número de restrições de tipo RO_3 violadas. Entretanto, esse processo computacional é muito mais demorado

4. Número de restrições de tipo RD_1 violadas:

A restrição de tipo RD_1 exige que um aluno não deve ser programado em mais de dois eventos consecutivos. Para encontrar o número de restrições RD_1 violadas, para cada aluno deve ser montado dois vetores auxiliares como o mostrado na Figura 5 para o aluno No. 1. Na verdade deve ser montado apenas o vetor que indica a hora programada para cada evento do aluno No. 1 já que os eventos se encontram na linha No. 1 da matriz C . Esse vetor de hora de programação do evento se encontra facilmente usando os dados de eventos do aluno (linha No. 1 da matriz C) e o vetor nho . Assim, para cada aluno pode ser montado o vetor auxiliar de horas mostrado na Figura 5.

Analisando o vetor de horas, pode-se facilmente determinar o número de restrições tipo RD_1 violadas para cada aluno. Assim, analisando o vetor de horas da Figura 5, pode-se verificar que o aluno No. 1 não viola restrições de tipo RD_1 (o aluno No. 1 tem aula nas horas 3, 6 e 8 no primeiro dia, nas horas 11, 14, 17 e 18 no segundo dia, nas horas 20, 23 e

Figura 5: Programação dos eventos do alunos No. 1.

24	35	50	101	161	175	231	243	248	257	268	273	279	283	324	364	374	Evento
28	31	24	23	41	14	6	20	38	17	3	11	8	34	43	18	33	Hora

Fonte: Elaborado pelo autor.

24 no terceiro dia, nas horas 28, 31, 33 e 34 no quarto dia e nas horas 38, 41 e 43 no último dia). Esse processo deve ser realizado para todos os alunos para determinar o número total de restrições do tipo RD_2 violadas.

5. Número de restrições de tipo RD_2 violadas:

A restrição de tipo RD_2 exige que um aluno não deve ser programado em apenas um evento em um dia. Para encontrar as restrições de tipo RD_2 violadas por cada aluno também pode ser usado o vetor de horas mostrado na Figura 5. Do vetor de horas mostrado na Figura 5, pode-se verificar que o aluno No. 1 não produz violações de restrições de tipo RD_2 (o aluno No. 1 tem 3 horas de aula no primeiro dia, 4 horas de aula no segundo dia, 3 horas de aula no terceiro dia, 4 horas de aula no quarto dia e 3 horas de aula no quinto dia). Esse processo deve ser realizado para todos os alunos para determinar o número total de restrições de tipo RD_2 violadas.

6. Número de restrições de tipo RD_3 violadas:

A restrição de tipo RD_3 exige que um aluno não deve ser programado no último horário do dia. Para encontrar as restrições de tipo RD_3 violadas por cada aluno também pode ser usado o vetor de horas mostrado na Figura 5. Assim, no vetor de horas de aula da Figura 5, deve-se verificar apenas se existem eventos programados nas horas 9, 18, 27, 36 e 45. No caso específico do aluno No. 1, mostrado na Figura 5, pode-se observar que existe uma violação da restrição RD_2 de parte do aluno No. 1 (o evento 364 está programado na hora 18 e faz parte da programação do aluno No. 1). Esse processo deve ser realizado para todos os alunos para determinar o número total de restrições de tipo RD_3 violadas.

Entretanto, existe uma forma mais rápida de determinar o número total de restrições de tipo RD_3 violadas. Para isso apenas precisa percorrer as linhas 9, 18, 27, 36 e 45 da matriz PG mostrada na Tabela 3 para identificar os eventos programados nessas horas. Como se conhece o número de alunos em cada evento, então de forma trivial pode ser encontrado o número total de restrições de tipo RD_3 violadas. Assim, para a proposta de solução apresentada na Tabela 3 (matriz PG) existem apenas 3 eventos programados na última hora de um dia (os eventos 8, 189 e 364). Assim apenas usando a informação do vetor b se encontra facilmente o número total de restrições de tipo RD_3 violadas. Neste caso específico esses dados são obtidos da Instância No. 1 do Apêndice (Número total de restrições de tipo RD_3 violadas: $0 + 4 + 3 = 7$ restrições violadas).

A forma sistemática apresentada anteriormente para calcular o número de restrições de cada tipo violadas pode ser implementada em qualquer tipo de meta-heurística. Essa estratégia foi usada em uma versão preliminar do algoritmo usado neste trabalho. Entretanto, as estruturas de vizinhança formuladas não apresentaram os resultados de desempenho esperados. Por esse

motivo neste trabalho não são apresentadas essas estruturas de vizinhança e nem a forma de implementação computacional.

Neste trabalho, as estruturas de vizinhança foram resolvidas resolvendo versões reduzidas do próprio modelo matemático apresentado anteriormente. Essa estratégia apresentou resultados de excelente qualidade em tempos de processamento não proibitivos. Adicionalmente, o modelo matemático apresentado anteriormente não pode ser usado para resolver o problema de otimização de grade horária para as instâncias analisadas neste trabalho porque o *solver* CPLEX não consegue resolver essas instâncias por problemas de memória para armazenamento de informação no processo de otimização.

5.4 Estrutura VNS Proposta

Na forma tradicional de implementar um algoritmo VNS para resolver um problema complexo normalmente são definidas as estruturas de vizinhança. Para cada nível de vizinhança, deve-se identificar o número de soluções vizinhas da solução corrente, deve-se verificar se essas propostas de solução vizinhas são factíveis, deve-se avaliar a qualidade de cada uma delas e finalmente as propostas de solução factíveis devem ser ordenadas em ordem de qualidade. Nesse contexto, por exemplo, um algoritmo VND identifica a melhor solução vizinha e se essa proposta de solução é melhor que a solução corrente. Nesse contexto merece um comentário adicional a forma em que se avalia a qualidade e a factibilidade de uma proposta de solução vizinha. Para mostrar os problemas de análise de vizinhança tradicional é usada a solução corrente mostrada na Tabela 3 da Seção 5.2. Supor que são formulados quatro tipos de vizinhança mostrados a seguir:

- Supor uma vizinhança, chamada de N_a , que define uma proposta de solução vizinha como sendo uma proposta encontrada a partir da solução corrente em que dois eventos programados na mesma sala trocam de horário. Esse tipo de vizinhança, embora seja muito simples, tem um número muito elevado de vizinhos. Assim, para a solução corrente mostrada na Tabela 3, a troca de dois eventos no horário dos eventos programados na sala 1 gera $41(40)/2 = 820$ vizinhos. Portanto, como existem 10 salas de aula o tamanho da vizinhança desse tipo, para alguma solução corrente, deve ser em torno de 8200 propostas de solução vizinhas. Nesse contexto, para cada uma das 8200 propostas de solução vizinhas devem ser verificadas e recalculadas as restrições de tipo RO_3 , RD_1 , RD_2 e RD_3 . É evidente o elevado esforço computacional para verificar a possibilidade de que dois eventos programados na mesma sala troquem de horário.
- Supor uma vizinhança, chamada de N_b , que define uma proposta de solução vizinha como sendo uma proposta encontrada a partir da solução corrente em que quatro pares de eventos, cada par de eventos programados na mesma sala, trocam de horário. Esse tipo de vizinhança, embora seja apenas uma extensão da vizinhança simples mostrada anteriormente, tem um número absurdo de vizinhos. Nesse contexto, para cada uma das propostas de solução vizinhas devem ser verificadas e recalculadas as restrições de tipo RO_3 , RD_1 , RD_2 e RD_3 . Também é evidente o elevado esforço computacional para verificar a possibilidade de que quatro pares de eventos, com cada par de eventos programados na mesma sala, troquem de horário.

- Supor uma vizinhança, chamada de N_c , que define uma proposta de solução vizinha como sendo uma proposta encontrada a partir da solução corrente da mesma forma em que foi definida no item anterior, mas os eventos também podem trocar de sala. Esse tipo de vizinhança é mais complexo em tamanho e na verificação da qualidade e da factibilidade. Nesse contexto, para cada uma das propostas de solução vizinhas devem ser verificadas e recalculadas as restrições de tipo RO_2 , RO_3 , RD_1 , RD_2 e RD_3 . Também é evidente o incremento do esforço computacional para verificar a possibilidade de que quatro pares de eventos, cada par programado na mesma sala, troquem de horário e de sala.
- Finalmente, supor uma vizinhança, chamada de N_d , que define uma proposta de solução vizinha como sendo qualquer proposta de solução que pode ser encontrada através de um remanejamento de todos os eventos programados em todas as salas e programadas no primeiro dia (horário de 1 a 9), isto é, todos os 80 eventos programados no primeiro dia podem ser remanejados o que equivale a desprogramar todos os eventos programados no primeiro dia e reprogramar novamente. Nesse caso o esforço computacional para identificar e avaliar cada proposta de solução vizinha é quase inviável de ser implementada na forma de identificação e avaliação de vizinhança tradicional. Nesse contexto, para cada uma das propostas de solução vizinhas devem ser verificadas e recalculadas as restrições de tipo RO_2 , RO_3 , RD_1 , RD_2 e RD_3 . Neste caso o incremento do esforço computacional é muito elevado. Uma alternativa para identificar a melhor solução vizinha da solução corrente pode ser obtida resolvendo o problema usando o modelo matemático apresentado anteriormente. Para implementar essa estratégia, deve-se fixar os valores de todos os eventos programados no intervalo horário de 10 a 45 e liberar as variáveis relacionadas com os eventos programados no horário 1 a 9. Em outras palavras, nesse caso, a estratégia consiste em resolver um subproblema do modelo matemático que é resolvido rapidamente e identifica a melhor solução vizinha de uma estrutura de vizinhança de elevada complexidade. Neste trabalho foi escolhido esse tipo de estratégia para identificação da melhor proposta de solução vizinha da solução corrente.

A estratégia de identificar a melhor solução vizinha da solução corrente, para um tipo de vizinhança escolhido, resolvendo um subproblema mais simples do próprio modelo matemático não é uma proposta totalmente inovadora. Entre as várias heurísticas existentes na literatura existe um tipo de heurística chamada de Heurística de Decomposição. Um subtipo dessa heurística é chamado de Heurística de Partição. Nesse tipo de heurística algumas variáveis podem ter seus valores fixados usando alguma estratégia interessante e, portanto, pode ser resolvido de forma exata o problema resultante que, obviamente, é de menor tamanho. Essa ideia fundamental é usada neste trabalho. Portanto, a proposta inovadora neste trabalho consiste em usar a Heurística de Partição para identificar a melhor solução vizinha da solução corrente após especificar a estrutura de vizinhança. Em resumo, neste trabalho a identificação da melhor solução vizinha é realizada resolvendo um problema de menor tamanho usando o próprio modelo matemático. O problema a ser resolvido é de menor tamanho porque uma grande parcela das variáveis binárias é fixada e o problema resultante é muito menor e mais fácil de resolver. O tipo de subproblema a ser resolvido depende do tipo de vizinhança em análise. Em todos os casos se resolve um problema da seguinte forma:

$$Min \ v = \sum_{a \in \Omega_A} \sum_{d=1}^5 (\alpha_3 f_{a,d} + \alpha_2 u_{a,d} + \alpha_1 \sum_{k=1}^7 c_{a,d,k}) + \beta \sum_{e \in \Omega_E} \Delta_e + \rho \sum_{e \in \Omega_E} \sum_{s \in \Omega_s} \varphi_{e,s} \quad (15)$$

s.a

$$\sum_{s \in \Omega_s} \sum_{t \in \Omega_t} x_{e,s,t} = 1 - \Delta_e \quad \forall e \in \Omega_E \quad (16)$$

$$\sum_{e \in \Omega_E} x_{e,s,t} \leq 1 \quad \forall s \in \Omega_s; \ t \in \Omega_t \quad (17)$$

$$x_{e,s,t} \leq ES_{e,s} + \varphi_{e,s} \quad \forall e \in \Omega_E; \ s \in \Omega_s; \ t \in \Omega_t \quad (18)$$

$$\varphi_{e,s} \leq 1 - ES_{e,s} \quad \forall e \in \Omega_E; \ s \in \Omega_s \quad (19)$$

$$HA_{a,t} = \sum_{e \in \Omega_E} EA_{e,a} \sum_{s \in \Omega_s} x_{e,s,t} \quad \forall a \in \Omega_A; \ t \in \Omega_t \quad (20)$$

$$HA_{a,t} \leq 1 \quad \forall a \in \Omega_A; \ t \in \Omega_t \quad (21)$$

$$\sum_{t=9d+k-9}^{9d+k-7} HA_{a,t} \leq 2 + c_{a,d,k} \quad \forall a \in \Omega_A; \ d = 1, 2, \dots, 5; \ k = 1, 2, \dots, 7 \quad (22)$$

$$f_{a,d} = HA_{a,9d} \quad \forall a \in \Omega_A; \ d = 1, 2, \dots, 5 \quad (23)$$

$$\sum_{t=9d-8}^{9d} HA_{a,t} = u_{a,d} + n_{a,d} \quad \forall a \in \Omega_A; \ d = 1, 2, \dots, 5 \quad (24)$$

$$\sum_{t=9d-8}^{9d} HA_{a,t} \geq 2\alpha_{a,d} \quad \forall a \in \Omega_A; \ d = 1, 2, \dots, 5 \quad (25)$$

$$2\alpha_{a,d} \leq n_{a,d} \leq 9\alpha_{a,d} \quad \forall a \in \Omega_A; \ d = 1, 2, \dots, 5 \quad (26)$$

$$\sum_{e \in \Omega_E} \sum_{s \in \Omega_s} \sum_{t \in \Omega_t | x_{e,s,t}^* = 0} (x_{e,s,t} - x_{e,s,t}^*) \leq \bar{K} \quad (27)$$

$$\sum_{e \in \Omega_E} \sum_{s \in \Omega_s} \sum_{t \in \Omega_t | x_{e,s,t}^* = 1} (x_{e,s,t}^* - x_{e,s,t}) \leq \bar{K} \quad (28)$$

$$\alpha_{a,d} \in \{0, 1\} \quad \forall a \in \Omega_A; \ d = 1, 2, \dots, 5 \quad (29)$$

$$x_{e,s,t} \in \{0, 1\} \quad \forall e \in \Omega_E; \ s \in \Omega_s; \ t \in \Omega_t \quad (30)$$

$$\Delta_e \in \{0, 1\} \quad \forall e \in \Omega_e \quad (31)$$

$$\varphi_{e,s} \in \{0, 1\} \quad \forall e \in \Omega_e; \ \forall s \in \Omega_s \quad (32)$$

$$HA_{a,t} \geq 0 \quad \forall a \in \Omega_a; \ \forall t \in \Omega_t \quad (33)$$

$$0 \leq c_{a,d,k} \leq 1 \quad \forall a \in \Omega_a; \ d = 1, \dots, 5; \ k = 1, 2, \dots, 7 \quad (34)$$

$$0 \leq f_{a,d} \leq 1 \quad \forall a \in \Omega_a; \ d = 1, \dots, 5 \quad (35)$$

$$0 \leq u_{a,d} \leq 1 \quad \forall a \in \Omega_a; \ d = 1, \dots, 5 \quad (36)$$

$$0 \leq n_{a,d} \leq 9 \quad \forall a \in \Omega_a; \ d = 1, \dots, 5 \quad (37)$$

O modelo matemático apresentado anteriormente é uma versão modificada do modelo matemático apresentado em (1)-(14). Na versão modificada a mudança mais significativa acontece

com a incorporação do parâmetro $x_{e,s,t}^*$ que identifica a solução corrente cuja vizinhança se pretende analisar e identificar a melhor solução vizinha, utilizando-se as restrições (27) e (20), de acordo com o parâmetro \bar{K} . Foram também adicionadas algumas grandezas adicionais e alguns parâmetros, assim como a conjunto Ω_t que indica o conjunto de intervalos de tempo que é igual a $t = 1, 2, \dots, 45$ no caso do problema analisado.

As variáveis Δ_e e $\varphi_{e,s}$ são introduzidas no modelo de forma a garantir que sempre se obtenha uma solução factível para o modelo modificado (que será infactível para o modelo original caso alguma dessas variáveis seja diferente de zero). Note que caso $\Delta_e = 1$ para um evento e , o evento é retirado da grade horária, e uma penalização é aplicada na função objetivo do problema. Da mesma forma, caso $\varphi_{e,s} = 1$, então um evento poderá ser alocado em uma sala que não cumpra com os requisitos do evento, e uma penalização ocorrerá na função objetivo. Os parâmetros α_1 , α_2 , α_3 , β e ρ são fatores de ponderação para penalidade de violação de cada tipo de restrição. Neste trabalho utilizou-se $\alpha_1 = 1$, $\alpha_2 = 1$, $\alpha_3 = 1$, $\beta = 1000$ e $\rho = 1000$.

Uma solução inicial para o problema é obtida fixando-se $\Delta_e = 1 \forall e \in \Omega_E$ e fazendo-se iterativamente $\Delta_e = 0$ de acordo com uma sequencia pré-determinada. Para cada mudança de $\Delta_e = 1$ para $\Delta_e = 0$, resolve-se o problema resultante e atualiza-se a solução corrente $x_{e,s,t}^*$, sendo que neste caso, $\bar{K} = 1$ no modelo. A ordem com que Δ_e é fixado em zero é do evento e com menos salas adequadas para o evento com mais salas adequadas. Caso fosse obtida uma solução com algum $\varphi_{e,s} = 1$, indicando que não foi possível alocar um evento em uma sala adequada, então uma etapa de reparação da solução em construção era realizada, permitindo-se a troca de até 3 eventos na grade horária ($\bar{K} = 3$).

Na implementação computacional foram usadas as seguintes estruturas de vizinhança para a etapa de melhoria local com o algoritmo VND na estrutura GVNS:

1. Estrutura N_1 :

Neste caso todos os eventos programados em uma sala podem ser trocados de horário. Usando a proposta de solução apresentada na Tabela 3, essa proposta de vizinhança equivale, por exemplo, a reordenar todos os 41 eventos programados na sala 1. Assim, todos os eventos programados na sala 1, podem ser reprogramados nas 45 horas de programação disponíveis. Desta forma, podem ser desprogramados e reprogramados todos os eventos programados em cada sala de aula. O modelo matemático encontra a melhor solução vizinha após resolver o modelo modificado para a proposta de solução corrente.

2. Estrutura N_2 :

Neste caso todos os eventos programados em duas salas podem ser trocados de horário. Usando a proposta de solução apresentada na Tabela 3, essa proposta de vizinhança equivale, por exemplo, a reordenar todos os 81 eventos programados nas salas 1 e 2. Assim, todos os eventos programados nas salas 1 e 2, podem ser reprogramados nas 90 horas de programação disponíveis. Desta forma, podem ser desprogramados e reprogramados todos os eventos programados em duas salas de aula simultaneamente. Como existem 10 salas de aula no caso típico apresentado na Tabela 3, então existem 45 estruturas de vizinhança diferentes desse tipo. O modelo matemático encontra a melhor solução vizinha após resolver o modelo modificado para a proposta de solução corrente.

3. Estrutura N_3 :

Neste caso foi permitido desprogramar um número máximo de eventos programados em 3 salas de aula. Assim, nos testes foi permitido desprogramar até 20 eventos ($\bar{K} = 20$) programados em 3 salas de aula. O modelo matemático modificado encontra a nova programação reprogramando até 20 eventos nas 3 salas de aula selecionadas. Deve-se observar que neste caso, para uma solução corrente, existem muitas propostas de solução vizinhas. Na verdade o número de soluções vizinhas cresce de forma significativa. O modelo matemático encontra a melhor solução vizinha após resolver o modelo modificado para a proposta de solução corrente.

4. Estrutura N_4 :

Neste caso foi permitido desprogramar um número máximo de eventos programados em um número de horários consecutivos. Assim, nos testes foram selecionados para desprogramar até 20 eventos ($\bar{K} = 20$) programados em 6 horários consecutivos. Por exemplo, na solução corrente mostrada na Tabela 3, podem ser desprogramados até 20 eventos programados nos horários de 1 a 6 (as seis primeiras horas do primeiro dia). O modelo matemático modificado encontra a nova programação reprogramando até 20 eventos nas seis horas consideradas para análise e ocupando os espaços liberados com a desprogramação de até 20 eventos e, eventualmente, usando espaços não usados na grade horária. Deve-se observar que neste caso, para uma solução corrente, também existem muitas propostas de solução vizinhas. Na verdade o número de soluções vizinhas cresce de forma ainda mais significativa. O modelo matemático encontra a melhor solução vizinha após resolver o modelo modificado para a proposta de solução corrente.

5. Estrutura N_5 :

Neste caso podem ser desprogramados um número máximo de eventos programados em um dia. Na implementação computacional, pretende-se encontrar a melhor solução vizinha encontrada mudando a programação de até 10 eventos ($\bar{K} = 10$) de um dia de programação. O modelo matemático encontra a melhor solução vizinha após resolver o modelo modificado para a proposta de solução corrente.

6. Estrutura N_6 :

Neste caso podem ser desprogramados um número máximo de eventos programados em dois dias. Na implementação computacional, pretende-se encontrar a melhor solução vizinha encontrada mudando a programação de até 5 eventos ($\bar{K} = 5$) em dois dias de programação. O modelo matemático encontra a melhor solução vizinha após resolver o modelo modificado para a proposta de solução corrente.

7. Estrutura N_7 :

Neste caso podem ser trocados até 3 eventos ($\bar{K} = 3$) da solução corrente. Assim, pretende-se encontrar a melhor solução vizinha em que 3 eventos podem mudar de posição dentro da grade horária (não existe restrição de deslocamento de sala nem de horário). O modelo matemático encontra a melhor solução vizinha após resolver o modelo modificado para a proposta de solução corrente.

8. Estrutura N_8 :

Neste caso foram desprogramados um número máximo de eventos da grade horária escolhidos de forma aleatória. Nos testes foram desprogramados até 20 eventos ($\bar{K} = 20$)

da grade horária, sendo que 60 % dos eventos são fixados na solução corrente de maneira aleatória. Assim, pretende-se encontrar a melhor solução vizinha quando são desprogramados até 20 eventos da grade horária dentre 40 % dos eventos que não estão fixados. O modelo matemático encontra a melhor solução vizinha após resolver o modelo modificado para a proposta de solução corrente.

Em todas as estruturas de vizinhança, os eventos que estão alocados na última hora do dia também são sempre liberados em cada vizinhança, bem como um número aleatório de eventos que não cumprem com a restrição desejável número 1 (neste caso, nas iterações iniciais do algoritmo, quando muitas violações ocorrem em RD_1 , a proporção de eventos que podem ser liberados que violam essa restrição deve ser pequena, podendo aumentar de acordo com a redução das violações dessa restrição).

Nos testes foram usadas as estruturas de vizinhança indicadas anteriormente dentro de uma estrutura de algoritmo GVNS. As vizinhanças N_1-N_8 são utilizadas na etapa de melhoria local realizada pelo algoritmo VND. Na etapa de agitação são desprogramados de 0 % a 100 % dos eventos da grade horária, com passos de 20 % em cada nível de vizinhança (os eventos são retirados da grade horária fixando-se o Δ_e correspondente em 1). Em seguida, a estratégia que constrói a solução inicial é utilizada para reprogramar as aulas que saíram.

Capítulo 6

Resultados de Testes

6.1 Introdução

Neste Capítulo são apresentados resultados de testes obtidos com o algoritmo VNS especializado implementado neste trabalho. São mostrados resultados de testes das 20 instâncias (20 problemas de grade horária diferentes) existentes no site do Concurso Internacional de Grade Horária. Nessas instâncias, mostrada na Tabela 4, podem ser observadas as seguintes características:

- O número de eventos varia de 350 a 440 eventos, mas a maioria das instâncias têm 400 eventos. Deve-se observar que para o caso em que existem 400 eventos e 450 possibilidades de programação de eventos (10 salas de aula e 45 horas de programação) é muito difícil encontrar soluções de excelente qualidade considerando que existem 50 horas que fazem parte da última hora de programação de aulas e nos quais a restrição secundária RD_3 não recomenda programar eventos. Portanto, uma programação ideal seria aquela que programe os 400 eventos em 400 espaços disponíveis na grade horária (sem levar em conta a última hora de programação de cada dia).
- Em 18 instâncias existem disponíveis 10 salas de aula e nas outras duas existem disponíveis 11 salas de aula.
- O número de alunos de cada instância varia de 200 a 350 alunos. Na verdade existem 6 instâncias com 200 alunos, 3 instâncias com 220 alunos, 2 instâncias com 250 alunos, 7 instâncias com 300 alunos e 2 instâncias com 350 alunos.
- Existem 5 dias de programação semanal da grade horária e 9 horas de programação por dia para programar cada evento de uma hora de duração para todas as instâncias.

Essas instâncias têm diferentes níveis de dificuldade. Um aspecto adicional relacionado com essas instâncias é que todas elas foram idealizadas de forma que a solução ótima de todas essas

Tabela 4: Instâncias dos problemas de grade horária.

Número da instância	Número de eventos	Número de salas	Número de alunos	Horas de aula por dia	Número de dias
1	400	10	200	9	5
2	400	10	200	9	5
3	400	10	200	9	5
4	400	10	300	9	5
5	350	10	300	9	5
6	350	10	300	9	5
7	350	10	350	9	5
8	400	10	250	9	5
9	440	11	220	9	5
10	400	10	200	9	5
11	400	10	220	9	5
12	400	10	200	9	5
13	400	10	250	9	5
14	350	10	350	9	5
15	350	10	300	9	5
16	440	11	220	9	5
17	350	10	300	9	5
18	400	10	200	9	5
19	400	10	300	9	5
20	350	10	300	9	5

Fonte: Elaborado pelo autor.

instâncias seja igual $v = 0$. Esse fato, dificulta ainda mais a solução dessas instâncias usando um modelo matemático exato e um *solver* de otimização como o CPLEX. Assim, quando é usado o CPLEX para resolver essas instâncias, em todos os testes, o CPLEX termina o processo de otimização abortando o processo por problemas de armazenamento de memória para armazenar a informação e mostrando a melhor solução encontrada e com a informação adicional de um *gap* igual a 100%. Assim, qualquer proposta de solução fornecida pelo CPLEX com um *gap* desse tipo indica uma solução de pobre qualidade.

Na Tabela 5 são mostrados os resultados encontrados pelo algoritmo GVNS desenvolvido neste trabalho. Na primeira coluna aparece a informação da instância, na segunda coluna aparece a função objetivo encontrado pelo algoritmo GVNS desenvolvido neste trabalho (número de restrições secundárias violadas). Nas quatro últimas colunas são mostrados os melhores resultados encontrados por outros pesquisadores. Na última linha da tabela é mostrada a soma das funções objetivo das 20 instâncias de cada proposta de otimização e que pode ser usada como medida de comparação dos resultados encontrados.

Dos resultados mostrados na Tabela 5 podem ser mencionados os seguintes fatos relevantes:

- Apenas uma proposta de otimização (a proposta 3) encontrou a solução ótima para uma

Tabela 5: Melhores resultados encontrados.

Número da instância	GVNS	1	2	3	4
1	40	45	61	85	63
2	23	25	39	42	46
3	56	65	77	84	96
4	136	115	160	119	166
5	166	102	161	77	203
6	48	13	42	6	92
7	14	44	52	12	118
8	29	29	54	32	66
9	33	17	50	184	51
10	61	61	72	90	81
11	41	44	53	73	65
12	84	107	110	79	119
13	76	78	109	91	160
14	74	52	93	36	197
15	40	24	62	27	114
16	21	22	34	300	38
17	119	86	114	79	212
18	16	31	38	39	40
19	97	44	128	86	185
20	13	7	26	0	17

Fonte: Elaborado pelo autor.

instância (a instância 20). Em todos os outros testes foram encontradas apenas soluções de boa qualidade (não ótimas).

- O algoritmo GVNS desenvolvido neste trabalho encontrou a melhor solução conhecida em 9 instâncias das 20 instâncias (nas instâncias 1, 2, 3, 8, 10, 11, 13, 16 e 18). A proposta de otimização 1 encontrou a melhor solução em seis casos (nas instâncias 4, 8, 9, 10, 15 e 19). A proposta de otimização 2 não encontrou a melhor solução em nenhuma das instâncias. A proposta de otimização 3 encontrou a melhor solução em 7 instâncias (nas instâncias 5, 6, 7, 12, 14, 17 e 20). A proposta de otimização 4 não encontrou a melhor solução em nenhuma das 20 instâncias. Finalmente, o algoritmo GVNS e a proposta de otimização 1 encontraram a melhor solução em duas instâncias (as instâncias 8 e 10). Essas soluções tem a mesma função objetivo, mas podem ser soluções ótimas alternativas.
- Se for considerado o número de instâncias com melhor desempenho, então o algoritmo GVNS desenvolvido neste trabalho apresentou melhor desempenho já que foi melhor que as outras técnicas de otimização em 7 casos e empatou em outros dois casos.
- Observando os valores das funções objetivo das 20 instâncias, pode-se concluir que a proposta de otimização 1 apresentou um bom desempenho. Pode-se verificar que essa proposta é mais estável, isto é, apresenta bom desempenho em todas as instâncias.

Nas Tabelas 6 a 25 são mostradas as grades horárias encontradas para as 20 instâncias. Deve-se observar que em quase todos os trabalhos de pesquisa relacionados com a área de pesquisa operacional não é comum apresentar os resultados completos, isto é, a grade horária encontrada neste caso. Na maioria dos casos são mostradas apenas os valores da função objetivo.

Finalmente, com a finalidade de verificação no Apêndice são mostrados os dados completos de algumas instâncias do problema de Competição Internacional de Grade Horária. Os dados completos de todas as instâncias podem ser encontrados no site da Competição Internacional de Grade Horária. Também, deve-se observar que o formato dos dados apresentados no Apêndice foram reformatados da forma mais adequada para realizar os testes usando o algoritmo GVNS.

Em relação aos quatro primeiros colocados no problema de competição de grade horária mostrados na Tabela 5 e obtidos do site da Competição Internacional de Otimização do Problema de Grade Horária, a seguir são mostrados alguns comentários dessas propostas de otimização obtidos do site citado.

O primeiro colocado da competição, com um total de 1011 violações nas restrições secundárias em todas as vinte instâncias utilizou um algoritmo de *Simulated Annealing* para resolver o problema. Uma fase de pré-processamento é aplicada para a obtenção da solução inicial que cumpre com as restrições obrigatórias. O tipo de vizinhança utilizada para reduzir o número de restrições desejáveis violadas considera a troca de dois eventos na grade horária. O grupo que ficou na segunda colocação da competição obteve soluções com um total de 1535 violações nas restrições desejáveis do problema usando um algoritmo de busca tabu. A estratégia proposta também é composta de uma etapa que obtém uma grade horária totalmente factível em relação às restrições obrigatórias, sendo que na segunda etapa o algoritmo de busca tabu minimiza o número de restrições desejáveis violadas. Consideram-se três tipos de movimentos no algoritmo proposto, isso é, troca de dois eventos de horário, troca de dois eventos em uma sala e troca de dois eventos de horário e de sala.

O terceiro colocado obteve soluções para o problema com um total de 1541 violações de restrições secundárias utilizando o *Great Deluge* (Grande Dilúvio) *local search algorithm*. A origem do nome desse algoritmo surge da analogia de uma pessoa escalando uma montanha fugindo de um dilúvio. A troca de dois eventos de sala e de horário é considerada como o tipo de vizinhança do algoritmo. A proposta da quarta colocação da competição obteve soluções com um total de 2129 restrições secundárias violadas usando um algoritmo de três estágios: (i) *Hill Climbing*, (ii) *Tabu Search* e (iii) Agitação. Na etapa (i) são considerados três tipos de vizinhança, sendo que a primeira troca o horário de um evento, sem modificar a sala, a segunda considera a troca de sala de um evento, sem modificar o horário e a terceira considera a troca de dois eventos de posição na grade horária. Na segunda etapa é utilizada uma memória tabu de tamanho variável e de curto prazo, e o tipo de vizinhança considerada é mover um evento na grade horária para um horário e em uma sala disponíveis. Na etapa de agitação, todos os eventos trocam de horário, mantendo a mesma sala.

Tabela 6: Melhor solução encontrada para a Instância No. 1.

Função objetivo: $v = 40$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	43	338	127	263	70	319	160	204	353	59
2	38	241	61	307	73	11	104	288	320	378
3	48	352	120	31	143	268	5	236	197	53
4	362	394	78	360	134	200	172	302	246	179
5	256	329	178	281	269	28	207	121	285	113
6	395	267	335	110	150	26	75	140	231	54
7	385	232	42	146	88	107	249	118	313	330
8	389	156	308	228	275	328	342	103	279	379
9	0	0	0	0	0	0	0	0	0	0
10	383	74	151	192	29	27	153	194	152	334
11	397	388	55	356	93	99	273	303	3	321
12	390	245	22	84	230	318	91	64	262	367
13	398	94	294	108	138	190	0	211	186	238
14	158	77	175	368	144	112	349	287	306	280
15	274	133	33	90	260	40	214	278	154	41
16	46	56	123	344	39	216	19	185	355	209
17	18	130	257	36	65	325	128	220	293	173
18	364	0	0	0	0	0	0	0	0	0
19	201	79	219	187	165	312	16	126	315	255
20	60	169	117	6	226	243	180	176	129	327
21	392	222	1	114	323	258	182	96	213	98
22	86	225	148	183	347	276	0	196	87	212
23	139	239	63	229	391	17	157	359	101	376
24	296	174	163	206	50	286	69	92	145	167
25	37	15	339	168	105	340	47	223	316	369
26	393	371	261	357	136	49	195	51	337	67
27	0	0	0	0	0	0	0	0	8	0
28	295	282	162	21	106	300	24	254	95	299
29	384	242	171	272	166	9	348	264	109	301
30	181	52	290	44	193	23	25	147	62	377
31	332	184	326	210	170	35	366	289	221	341
32	132	141	97	12	122	66	265	68	331	358
33	363	333	45	32	374	322	102	0	346	336
34	387	198	283	4	76	250	58	247	135	224
35	89	343	297	149	400	270	155	234	350	380
36	0	0	0	0	0	0	0	0	0	0
37	396	309	271	372	203	284	298	314	305	240
38	248	215	82	304	375	34	266	218	205	351
39	365	115	252	81	30	131	345	233	125	370
40	57	199	159	259	13	217	227	208	361	164
41	191	137	7	124	116	202	161	111	177	235
42	85	311	354	244	399	14	119	291	317	381
43	2	237	20	71	386	292	83	72	324	188
44	80	382	251	253	10	310	142	100	277	373
45	0	0	0	0	189	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 7: Melhor solução encontrada para a Instância No. 2.

Função objetivo: $v = 23$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	187	219	151	206	316	152	361	299	38	29
2	346	261	186	262	387	392	54	284	301	396
3	73	228	101	110	176	226	148	127	172	239
4	327	285	279	379	383	5	253	268	143	170
5	65	19	160	215	40	348	227	270	300	295
6	83	132	378	30	382	391	70	171	247	8
7	130	280	154	113	340	145	223	56	294	224
8	120	276	91	205	312	394	342	140	88	17
9	0	0	0	0	69	0	0	0	0	0
10	81	4	203	58	384	305	199	240	208	231
11	157	352	380	94	339	289	104	173	78	144
12	243	366	61	317	385	356	212	302	345	278
13	45	121	372	246	179	158	167	114	142	277
14	82	343	260	293	281	183	221	198	13	153
15	296	146	373	139	254	217	141	308	68	0
16	184	324	255	337	322	34	204	16	220	3
17	311	272	368	376	235	273	195	256	355	398
18	0	0	0	0	0	0	0	0	0	0
19	304	298	344	381	28	117	400	286	55	354
20	193	309	371	269	135	230	248	71	67	62
21	36	92	133	129	245	395	93	303	292	97
22	9	367	25	364	359	74	177	2	107	290
23	57	20	282	331	237	397	1	315	169	27
24	209	328	75	72	288	118	123	77	41	134
25	32	287	109	375	271	347	214	335	80	341
26	330	307	241	266	90	201	64	166	232	102
27	0	47	0	0	0	0	0	0	0	0
28	377	43	147	238	182	390	162	116	100	225
29	76	207	258	249	369	244	26	334	89	350
30	363	119	326	175	15	358	159	196	178	200
31	98	0	59	0	156	353	95	320	263	49
32	48	318	86	96	388	111	87	313	23	336
33	108	333	211	188	60	393	242	51	319	233
34	22	191	265	122	229	24	44	323	168	46
35	306	105	332	106	202	360	35	218	115	112
36	0	234	0	0	192	0	0	0	0	0
37	128	42	63	31	386	52	21	374	53	14
38	197	365	136	131	185	99	251	10	210	190
39	264	138	236	310	126	349	50	79	329	0
40	321	189	370	39	164	149	155	325	84	213
41	362	274	103	259	163	7	399	85	291	37
42	161	174	150	338	18	11	33	124	216	357
43	275	165	66	314	389	252	6	297	137	194
44	250	222	181	180	351	12	257	267	283	125
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 8: Melhor solução encontrada para a Instância No. 3.

Função objetivo: $v = 56$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	238	372	234	223	384	45	28	246	18	81
2	255	242	210	20	15	46	300	353	125	375
3	366	5	201	58	38	96	60	176	168	70
4	389	357	72	156	177	62	368	167	274	303
5	391	197	250	335	363	217	14	115	345	251
6	13	68	3	173	80	0	148	241	155	83
7	243	33	228	59	339	159	340	219	371	379
8	305	397	227	85	281	316	309	307	100	11
9	0	0	0	355	0	0	0	0	0	0
10	191	225	130	240	338	39	10	192	129	86
11	170	302	63	117	215	0	308	268	135	313
12	180	19	330	43	295	116	320	349	137	169
13	390	350	88	151	327	54	61	207	224	0
14	211	249	361	283	56	95	292	1	202	165
15	138	317	111	254	298	50	392	245	297	383
16	244	161	214	179	174	314	187	90	188	0
17	92	400	333	319	47	278	344	277	118	291
18	0	0	0	306	0	0	0	0	0	0
19	336	354	91	220	377	280	289	157	360	323
20	127	394	259	315	282	212	34	205	162	285
21	272	247	76	101	4	385	74	123	190	24
22	106	290	108	94	221	326	44	105	141	318
23	131	150	382	126	128	386	296	16	370	376
24	387	237	218	142	351	30	55	103	79	230
25	52	239	209	189	199	97	120	109	342	119
26	358	153	264	147	112	226	134	75	236	32
27	0	0	0	149	57	0	0	0	0	0
28	299	107	2	262	252	232	89	9	367	263
29	84	257	248	195	312	362	98	258	35	399
30	393	398	310	166	93	381	42	267	110	7
31	322	233	26	321	64	304	87	329	204	279
32	122	334	152	216	17	184	346	271	144	145
33	332	146	36	198	185	113	347	294	337	359
34	182	124	288	183	12	328	73	193	269	373
35	256	395	41	208	222	49	365	6	293	178
36	0	0	0	0	0	0	0	0	0	0
37	114	273	276	65	364	261	121	352	53	203
38	270	27	266	158	194	163	348	301	31	78
39	171	311	253	175	133	229	206	284	235	154
40	341	200	396	77	369	378	343	48	275	102
41	172	37	265	331	29	164	51	104	71	82
42	356	139	132	21	67	380	213	374	181	136
43	25	260	99	325	22	388	8	286	186	231
44	69	140	23	196	66	324	160	40	287	143
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 9: Melhor solução encontrada para a Instância No. 4.

Função objetivo: $v = 136$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	179	4	280	372	85	9	141	6	399	205
2	176	210	352	52	251	253	71	292	196	148
3	157	99	356	348	56	250	222	3	369	121
4	15	239	82	261	325	19	236	65	335	271
5	155	180	51	226	133	197	353	18	10	296
6	391	39	233	324	323	216	388	327	36	252
7	151	103	315	365	189	177	337	38	160	69
8	144	168	79	20	49	66	338	161	153	392
9	0	0	0	0	0	0	0	0	0	0
10	360	5	339	26	137	140	12	147	303	30
11	123	366	248	272	21	310	107	13	286	81
12	115	308	73	48	396	122	390	257	314	386
13	240	342	203	75	184	0	132	336	378	330
14	213	363	116	64	29	129	95	198	393	175
15	114	362	232	361	28	97	201	259	384	68
16	152	170	190	207	194	93	246	108	91	139
17	212	72	263	221	7	281	349	288	345	172
18	0	0	0	0	0	0	0	313	0	0
19	42	181	368	78	283	46	351	297	106	211
20	74	377	225	249	35	128	234	178	385	238
21	266	156	40	163	282	316	200	22	358	320
22	302	245	228	284	105	332	70	193	113	340
23	117	62	63	247	215	45	167	149	146	199
24	291	41	169	398	269	319	273	347	329	127
25	1	187	285	218	214	119	17	34	367	231
26	206	23	267	33	204	154	387	304	183	192
27	0	124	0	0	0	0	0	57	0	0
28	258	134	260	61	158	2	268	8	150	24
29	298	125	290	293	242	185	118	44	294	202
30	400	55	92	374	165	274	120	89	131	279
31	142	98	191	277	309	143	295	305	350	27
32	209	328	138	25	220	11	60	0	126	346
33	14	173	270	370	278	37	0	162	357	84
34	317	223	136	110	306	159	265	90	382	32
35	208	100	50	375	379	96	237	164	101	376
36	0	0	0	0	0	0	0	227	0	0
37	256	53	364	341	235	381	371	80	241	359
38	331	322	67	229	87	76	321	0	264	135
39	219	43	217	174	16	86	380	230	394	289
40	243	102	188	334	112	244	354	312	397	145
41	186	94	31	373	344	299	130	182	58	224
42	166	333	301	307	311	318	104	54	383	109
43	83	287	326	355	88	77	389	47	255	254
44	59	111	195	300	276	171	343	262	395	275
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 10: Melhor solução encontrada para a Instância No. 5.

Função objetivo: $v = 166$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	316	0	22	3	193	68	107	266	0	350
2	47	190	133	86	5	183	103	293	233	87
3	289	229	337	222	200	168	89	32	251	153
4	31	333	36	115	8	118	33	16	0	339
5	128	66	0	34	88	57	72	334	0	101
6	197	201	129	314	113	300	0	281	0	245
7	6	346	172	28	134	159	217	110	0	276
8	320	21	270	1	165	313	184	310	55	156
9	0	0	0	0	0	0	0	0	0	0
10	188	235	0	274	18	38	19	9	216	41
11	79	282	26	73	331	23	105	95	273	223
12	0	0	344	108	141	252	0	24	0	0
13	39	49	307	52	347	50	10	98	0	83
14	45	209	30	295	349	202	0	131	0	304
15	264	2	0	226	121	140	144	214	0	174
16	303	61	232	210	158	258	279	136	14	58
17	119	312	284	124	315	149	311	65	120	345
18	0	0	0	0	0	0	0	0	0	0
19	243	227	147	224	186	286	205	20	0	246
20	0	0	0	234	91	288	160	151	275	206
21	296	0	37	327	17	0	265	299	0	162
22	44	51	0	145	263	142	204	317	259	328
23	248	194	319	97	0	297	85	228	0	198
24	283	0	0	75	238	294	0	231	225	342
25	150	0	326	249	219	99	277	148	254	268
26	63	139	0	207	185	84	106	199	53	122
27	0	0	0	0	0	0	0	0	0	0
28	167	247	298	59	239	196	211	25	130	309
29	250	117	11	301	340	241	179	62	176	330
30	146	290	166	242	182	13	0	329	0	322
31	220	280	56	64	338	169	178	48	257	318
32	305	0	187	203	218	287	302	42	212	29
33	0	171	82	143	306	125	191	12	0	0
34	195	43	0	116	267	94	109	336	0	292
35	324	155	70	240	341	260	213	332	135	343
36	0	0	0	0	0	0	0	0	0	0
37	261	262	60	272	177	164	54	78	271	308
38	112	0	0	127	76	236	321	15	154	27
39	173	7	291	181	256	215	237	269	0	0
40	69	92	253	230	138	123	0	192	137	163
41	114	189	0	175	161	104	152	323	244	102
42	132	0	74	80	81	255	278	208	0	325
43	93	170	335	67	285	4	100	180	157	126
44	348	71	96	46	111	77	35	90	221	40
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 11: Melhor solução encontrada para a Instância No. 6.

Função objetivo: $v = 48$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	156	211	237	163	39	19	33	334	0	68
2	176	219	108	85	0	37	267	51	137	297
3	2	311	70	0	0	90	343	168	0	0
4	0	266	340	27	0	346	207	117	190	38
5	235	0	86	54	155	289	201	46	132	104
6	0	102	157	164	298	147	36	0	110	65
7	167	295	221	265	276	316	67	347	174	271
8	172	198	89	223	5	105	256	188	127	58
9	0	0	0	0	0	239	0	0	0	0
10	185	100	253	244	120	4	48	278	228	125
11	26	326	106	241	187	148	87	281	182	130
12	0	0	8	113	280	327	332	214	0	0
13	302	212	9	318	0	262	66	285	98	303
14	234	144	32	279	310	20	57	150	0	0
15	53	0	299	308	0	288	153	97	0	23
16	0	290	246	194	17	335	84	11	255	317
17	56	258	129	136	204	55	301	224	230	330
18	0	0	0	0	0	0	209	0	0	0
19	138	116	254	149	142	220	215	189	14	179
20	249	15	260	341	203	275	175	61	107	119
21	52	238	222	315	261	0	277	0	0	0
22	273	0	231	45	218	293	140	339	103	60
23	0	80	349	250	79	282	184	304	92	114
24	292	248	94	21	0	208	336	286	0	200
25	81	307	43	135	62	35	284	161	345	47
26	25	152	226	143	245	31	191	29	170	69
27	0	0	0	0	74	0	0	0	0	0
28	233	64	96	305	95	123	350	309	91	320
29	112	0	259	333	331	159	63	173	141	199
30	1	16	30	139	134	217	0	165	0	0
31	263	0	257	40	242	18	197	0	0	322
32	50	0	348	294	300	225	306	34	71	205
33	0	227	272	76	3	162	210	0	121	59
34	75	319	99	296	252	247	160	0	291	0
35	328	329	154	109	93	236	314	78	181	313
36	0	0	0	0	0	0	0	0	0	0
37	118	146	269	22	122	6	101	287	169	88
38	202	178	73	240	251	42	325	124	324	0
39	243	0	180	77	115	28	195	0	0	323
40	344	0	44	274	177	193	10	111	0	338
41	166	337	264	206	216	49	213	186	270	0
42	7	0	13	128	232	131	283	0	183	312
43	24	0	41	145	342	83	171	229	12	158
44	72	192	133	321	196	82	151	126	268	0
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 12: Melhor solução encontrada para a Instância No. 7.

Função objetivo: $v = 14$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	141	318	48	245	326	135	201	231	1	69
2	0	67	324	58	229	208	163	36	73	199
3	124	332	214	281	0	0	0	0	11	176
4	0	0	0	228	15	184	0	0	137	46
5	256	47	49	0	218	60	284	251	294	151
6	235	114	51	174	319	30	0	89	274	259
7	321	0	154	193	0	271	0	322	32	270
8	75	0	160	248	197	150	288	99	5	87
9	0	0	0	0	0	0	0	0	0	182
10	64	194	349	108	216	128	168	27	232	74
11	215	0	50	125	44	230	107	293	305	101
12	0	70	209	348	96	297	278	246	34	0
13	238	330	203	296	196	126	0	0	340	0
14	299	240	204	213	180	170	266	147	222	102
15	4	0	119	25	76	345	0	273	90	23
16	39	333	68	277	308	98	301	275	139	148
17	311	13	350	310	7	290	300	6	122	181
18	0	0	0	0	0	195	0	0	0	0
19	104	130	192	190	187	78	334	61	144	86
20	93	320	227	249	12	153	276	258	226	100
21	0	0	80	0	260	225	9	0	97	312
22	83	223	188	129	35	71	189	0	325	263
23	155	164	177	45	161	346	252	127	255	191
24	0	43	8	0	0	298	63	244	56	162
25	173	112	247	186	304	29	257	0	116	328
26	152	26	143	31	94	342	131	217	115	55
27	0	0	0	0	0	0	0	0	0	178
28	243	253	172	202	40	315	242	41	237	262
29	241	82	109	53	133	179	205	267	254	2
30	279	0	72	123	17	85	157	103	138	20
31	132	136	140	337	145	117	0	183	219	21
32	0	0	347	134	309	233	0	37	19	200
33	0	0	0	0	344	52	33	84	54	105
34	224	0	66	341	221	250	220	0	268	16
35	265	149	264	106	339	118	285	142	57	289
36	0	0	0	338	0	0	0	0	0	0
37	198	0	343	280	327	210	211	91	166	302
38	0	156	314	234	146	261	110	303	18	272
39	306	0	3	329	236	22	295	323	316	269
40	292	0	111	0	0	0	165	169	77	38
41	331	0	65	336	291	14	307	282	335	10
42	0	0	24	79	212	88	62	283	95	313
43	121	0	28	317	207	81	239	167	287	175
44	206	159	185	42	120	59	92	286	113	171
45	0	0	0	0	0	0	158	0	0	0

Fonte: Elaborado pelo autor.

Tabela 13: Melhor solução encontrada para a Instância No. 8.

Função objetivo: $v = 29$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	306	133	69	101	163	175	264	284	388	379
2	76	33	5	332	254	178	286	387	149	205
3	277	316	63	97	221	160	239	347	258	333
4	281	85	395	109	351	375	186	224	151	22
5	123	172	194	70	0	381	257	249	244	132
6	21	115	13	187	0	32	166	147	53	218
7	82	17	278	137	153	331	103	12	394	129
8	337	14	83	36	145	179	312	338	39	81
9	0	0	0	0	0	0	0	0	0	40
10	400	189	309	197	200	15	217	6	210	195
11	215	0	37	177	233	191	274	376	321	38
12	307	61	260	339	373	18	282	358	203	382
13	19	314	170	219	0	0	246	256	107	11
14	344	122	68	353	57	89	236	340	134	140
15	190	211	295	142	106	355	94	158	335	183
16	198	214	292	253	100	25	144	141	86	79
17	272	216	346	259	226	269	121	162	389	229
18	0	0	0	0	0	0	0	0	0	0
19	291	196	326	237	350	225	315	28	235	267
20	84	143	181	87	368	193	313	243	391	209
21	208	182	16	9	48	155	74	377	114	293
22	366	324	4	30	45	49	10	360	397	92
23	130	125	280	273	248	164	91	207	361	343
24	348	139	168	64	261	383	230	42	99	270
25	311	96	113	364	0	128	241	386	220	43
26	98	8	77	118	372	288	390	223	161	399
27	34	0	0	0	0	0	0	0	0	0
28	80	247	95	111	371	1	345	120	231	67
29	206	263	238	367	369	150	2	342	359	185
30	159	265	202	73	283	104	300	52	146	165
31	72	0	116	303	3	135	131	126	271	35
32	228	319	55	174	325	322	374	352	173	124
33	47	287	285	336	0	136	329	308	156	213
34	88	157	310	290	24	357	234	255	393	298
35	93	112	328	199	152	354	341	384	245	252
36	227	50	0	0	59	0	0	0	0	380
37	138	251	266	23	275	20	349	51	71	317
38	78	117	110	302	330	378	262	289	299	362
39	297	240	180	176	242	365	102	7	396	188
40	127	66	320	327	268	60	90	222	392	192
41	184	27	279	41	148	65	204	167	318	301
42	29	232	304	119	276	201	169	385	212	296
43	356	294	334	154	370	105	58	323	56	398
44	26	54	75	305	108	363	171	44	31	46
45	0	0	0	250	0	0	0	0	0	62

Fonte: Elaborado pelo autor.

Tabela 14: Melhor solução encontrada para a Instância No. 9.

Função objetivo: $v = 33$

Hora	Salas de aula										
	1	2	3	4	5	6	7	8	9	10	11
1	188	164	195	316	12	281	29	67	104	92	296
2	383	75	194	340	350	163	105	259	229	253	363
3	156	42	427	181	245	56	248	375	171	429	213
4	324	203	415	332	377	51	230	179	134	226	0
5	17	38	102	166	172	299	16	311	301	342	400
6	437	158	201	298	54	133	45	32	258	190	419
7	356	94	431	335	334	251	331	193	129	112	97
8	3	382	10	361	378	434	381	310	236	373	262
9	0	401	0	0	0	0	0	0	0	0	0
10	211	215	387	321	364	235	394	33	1	295	293
11	135	66	81	20	99	169	130	14	270	200	268
12	238	79	2	338	196	413	313	317	68	154	220
13	286	323	214	255	339	90	232	183	309	161	297
14	27	109	76	167	404	348	440	244	341	115	197
15	428	228	11	284	132	302	274	407	333	399	366
16	74	242	122	264	184	305	260	287	417	435	388
17	227	71	223	139	397	157	352	408	175	426	116
18	0	0	0	0	0	0	30	0	0	0	0
19	392	70	322	58	23	359	300	290	239	121	60
20	146	107	422	24	351	59	368	150	424	218	206
21	438	307	345	315	369	128	43	85	108	5	406
22	374	328	380	187	346	111	319	53	336	209	385
23	277	83	425	6	269	208	272	138	26	355	327
24	22	173	34	31	337	384	73	44	153	217	414
25	191	28	205	246	365	100	314	177	117	93	80
26	285	151	40	224	47	103	210	86	282	46	353
27	0	0	0	0	0	0	0	0	0	0	0
28	216	55	405	142	257	101	0	36	278	219	289
29	127	147	91	303	143	198	247	252	329	294	412
30	174	389	125	84	288	186	145	357	254	212	360
31	273	222	349	95	0	63	110	180	118	37	78
32	386	189	420	280	165	57	423	243	390	39	8
33	402	89	21	318	185	234	72	275	343	140	64
34	430	123	391	170	379	416	279	233	292	136	308
35	199	291	411	250	106	120	283	261	77	148	409
36	0	0	0	347	0	0	0	13	0	0	0
37	433	354	325	18	98	418	113	370	204	207	15
38	304	358	396	52	35	124	267	25	7	320	49
39	432	155	0	48	312	137	326	256	141	65	371
40	131	306	393	221	398	82	421	159	403	160	410
41	240	69	176	271	372	265	202	178	88	114	395
42	41	225	192	168	231	241	4	344	237	19	62
43	162	126	249	144	0	50	276	119	376	330	182
44	152	61	96	9	263	87	436	367	266	149	362
45	0	0	0	0	0	0	439	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 15: Melhor solução encontrada para a Instância No. 10.

Função objetivo: $v = 61$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	218	135	52	334	354	215	1	273	3	326
2	382	130	213	136	293	237	186	270	294	255
3	217	316	191	91	265	188	0	0	336	302
4	288	129	350	266	342	169	147	178	39	236
5	170	96	329	56	321	357	301	103	179	72
6	376	303	280	351	297	233	106	92	231	307
7	26	311	131	82	11	10	242	292	64	51
8	234	368	42	60	283	244	140	6	375	304
9	0	0	0	0	287	0	0	0	125	0
10	359	369	71	377	159	145	239	9	150	63
11	305	349	177	149	388	248	13	18	66	162
12	81	291	87	393	108	394	274	79	25	146
13	0	341	203	144	232	229	386	345	187	154
14	0	230	157	155	148	262	246	27	272	252
15	121	261	387	202	352	48	331	137	181	328
16	172	205	216	153	118	89	396	385	306	128
17	260	226	398	59	347	259	75	166	247	348
18	0	0	0	0	0	0	0	0	175	0
19	16	24	254	55	390	29	123	207	298	7
20	143	69	399	119	94	367	373	31	44	263
21	0	173	211	182	40	117	200	93	275	113
22	0	171	101	361	105	269	164	112	221	335
23	282	256	310	127	284	192	185	258	152	111
24	379	80	208	50	333	317	201	76	338	90
25	116	332	20	204	228	366	241	295	58	43
26	193	34	245	115	151	14	300	38	107	99
27	0	257	0	0	0	0	0	0	225	0
28	327	138	165	380	168	206	397	384	279	74
29	360	286	21	141	392	299	322	98	68	330
30	224	49	199	134	323	23	36	250	46	47
31	198	314	378	32	33	35	372	210	120	340
32	160	212	78	389	41	374	102	110	238	57
33	176	167	37	109	243	395	240	267	5	88
34	19	319	268	183	222	253	362	142	196	86
35	312	12	264	391	249	276	67	104	355	308
36	0	0	0	0	0	0	0	0	195	0
37	381	85	17	364	209	84	174	309	315	320
38	371	325	161	219	339	70	281	114	132	73
39	346	235	251	97	83	189	2	77	30	318
40	0	356	194	163	344	95	122	158	61	313
41	197	156	45	343	190	271	28	133	353	277
42	54	363	400	62	65	0	383	227	124	214
43	324	8	289	370	22	296	337	365	285	4
44	53	139	180	220	126	184	223	290	100	15
45	0	0	0	0	0	0	0	0	278	358

Fonte: Elaborado pelo autor.

Tabela 16: Melhor solução encontrada para a Instância No. 11.

Função objetivo: $v = 41$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	371	267	107	149	336	158	77	18	100	160
2	256	397	88	245	302	354	263	128	304	89
3	0	365	252	46	330	368	22	233	390	190
4	139	137	265	318	317	301	81	169	130	40
5	56	97	235	172	266	177	42	61	349	254
6	246	0	283	144	217	369	123	292	359	71
7	2	280	253	299	239	228	238	178	162	32
8	351	279	126	227	293	16	387	260	222	398
9	0	0	23	0	0	0	112	154	0	0
10	200	207	182	58	274	356	388	41	364	170
11	363	64	231	148	76	10	121	287	295	297
12	0	367	310	127	183	187	385	157	350	0
13	164	86	241	34	335	188	383	202	377	236
14	276	53	210	84	0	272	5	311	199	50
15	326	362	106	33	135	102	361	209	151	93
16	286	141	247	294	352	91	212	9	8	37
17	255	257	99	156	122	218	214	201	342	230
18	0	0	0	0	0	0	379	0	0	0
19	47	35	277	181	244	133	83	66	392	94
20	249	161	57	234	95	251	215	262	213	21
21	220	138	223	13	24	38	360	143	258	0
22	0	374	59	124	240	370	4	268	193	243
23	353	211	36	357	159	51	391	229	145	111
24	281	166	306	296	323	105	186	165	372	174
25	333	305	20	70	307	1	312	320	225	108
26	62	389	198	96	15	175	146	167	289	376
27	0	0	0	43	0	0	185	0	0	0
28	282	394	28	136	132	208	68	191	340	206
29	80	180	55	308	29	375	382	48	27	399
30	355	248	322	224	45	242	197	329	90	303
31	259	316	63	396	60	150	153	125	3	344
32	315	393	189	300	147	216	341	232	114	395
33	176	0	129	103	19	237	49	104	72	85
34	226	25	11	152	325	328	334	192	194	195
35	120	346	131	358	309	380	273	203	67	0
36	0	0	0	0	0	0	115	0	119	0
37	98	373	291	173	155	343	386	44	319	142
38	298	17	87	285	324	314	384	30	113	12
39	140	347	221	163	332	117	338	196	74	39
40	31	54	65	261	171	270	134	52	101	378
41	327	400	219	269	82	75	26	284	288	79
42	321	14	271	116	264	339	118	184	73	110
43	205	381	275	290	345	7	92	313	179	6
44	337	348	204	78	109	250	331	69	366	168
45	0	0	0	278	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 17: Melhor solução encontrada para a Instância No. 12.

Função objetivo: $v = 84$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	53	86	332	137	208	106	24	19	368	260
2	320	261	231	13	356	113	103	245	23	219
3	306	147	240	74	44	92	50	227	87	186
4	0	121	364	272	362	212	270	363	30	119
5	322	293	110	379	38	337	176	7	122	217
6	300	265	155	259	333	49	378	2	358	132
7	149	213	39	291	25	361	391	104	312	251
8	342	63	298	82	384	126	330	233	18	218
9	281	0	0	0	0	241	0	0	393	0
10	26	188	52	4	71	21	141	181	11	112
11	239	136	151	178	76	389	120	58	195	91
12	279	308	366	374	284	222	323	207	129	256
13	139	192	201	250	354	390	211	78	350	179
14	285	296	1	80	341	114	234	0	224	174
15	314	360	236	221	385	396	184	89	45	203
16	316	346	349	146	54	376	353	164	65	142
17	324	128	248	375	34	130	252	220	197	166
18	0	0	0	0	0	37	0	0	294	0
19	326	20	344	107	290	387	230	177	335	36
20	345	108	277	295	266	134	309	357	288	56
21	339	348	304	373	249	327	99	301	131	64
22	328	367	102	381	133	365	98	253	204	83
23	160	148	170	215	317	187	398	140	29	247
24	32	0	16	48	355	321	199	172	27	66
25	206	191	115	180	280	238	193	57	84	47
26	59	214	271	157	42	394	31	55	154	17
27	0	0	0	77	0	0	0	0	152	124
28	90	243	135	51	386	395	95	61	123	117
29	340	307	297	73	336	359	263	0	72	257
30	286	287	12	105	242	109	85	46	196	101
31	273	0	182	111	382	33	81	274	67	246
32	6	159	22	96	169	185	338	161	268	194
33	262	255	371	163	228	167	62	144	116	237
34	334	302	310	205	35	173	200	305	127	183
35	165	267	156	315	15	175	3	70	138	150
36	0	0	0	0	0	0	0	0	0	388
37	292	351	143	380	319	311	235	269	244	223
38	79	347	352	41	125	377	171	0	153	209
39	282	370	372	303	118	0	14	283	198	229
40	0	9	318	158	145	392	28	75	210	68
41	94	225	226	343	254	0	0	400	8	69
42	202	369	329	383	190	0	397	399	189	97
43	299	60	232	331	313	278	10	264	100	168
44	162	93	325	258	216	0	43	5	289	40
45	0	0	0	0	275	276	88	0	0	0

Fonte: Elaborado pelo autor.

Tabela 18: Melhor solução encontrada para a Instância No. 13.

Função objetivo: $v = 76$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	281	76	267	330	128	48	278	32	218	5
2	399	248	371	226	326	150	223	335	31	386
3	206	190	142	93	25	107	262	360	252	38
4	324	151	126	322	240	132	202	340	276	338
5	359	21	222	311	77	15	60	61	325	102
6	105	349	22	154	200	62	304	0	365	297
7	24	66	256	336	183	109	264	239	171	41
8	47	87	180	282	63	279	310	158	97	230
9	0	364	0	153	68	0	0	0	0	0
10	318	118	8	346	145	333	161	261	129	196
11	307	125	241	103	162	92	72	376	123	298
12	83	16	19	344	45	159	321	358	112	299
13	168	242	302	191	157	260	263	268	385	52
14	397	208	121	272	94	269	197	89	80	369
15	396	91	172	288	317	58	81	350	137	9
16	27	300	6	236	209	347	53	20	33	305
17	383	43	40	309	235	246	255	357	133	287
18	382	0	0	0	0	0	0	56	0	0
19	392	329	250	342	86	96	49	380	114	388
20	395	194	127	110	35	143	294	82	211	363
21	176	331	224	170	59	341	74	379	13	185
22	348	249	99	274	101	319	166	247	232	78
23	393	339	0	67	213	334	220	292	323	238
24	391	7	291	187	139	122	50	301	373	387
25	394	46	273	229	39	280	296	3	312	167
26	193	182	55	234	195	36	34	98	203	370
27	0	186	0	0	141	0	0	0	0	0
28	119	18	377	65	177	79	44	131	337	17
29	106	289	245	100	69	355	111	306	130	51
30	160	1	315	265	217	28	293	214	343	88
31	75	57	332	165	148	64	124	115	210	283
32	104	366	30	138	140	37	0	285	205	113
33	284	362	0	181	375	12	14	367	253	219
34	389	29	228	178	227	90	136	0	225	144
35	400	215	258	147	257	356	42	374	328	361
36	0	0	0	0	0	0	0	0	0	0
37	204	0	271	71	233	345	26	0	384	117
38	4	0	390	184	135	11	295	372	54	155
39	237	0	179	189	199	120	266	275	192	378
40	381	216	201	351	316	70	327	85	188	303
41	108	164	290	259	10	313	243	198	173	174
42	146	277	0	23	231	254	2	368	286	84
43	398	149	207	320	163	314	169	152	95	308
44	251	175	270	156	212	354	353	352	221	134
45	0	244	0	0	73	0	0	0	0	116

Fonte: Elaborado pelo autor.

Tabela 19: Melhor solução encontrada para a Instância No. 14.

Função objetivo: $v = 74$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	8	282	329	135	4	239	271	304	307	312
2	206	188	323	0	154	47	334	100	330	279
3	165	38	169	0	0	0	9	113	190	234
4	270	77	54	0	36	0	30	74	0	0
5	63	15	131	335	20	216	283	343	189	111
6	248	128	91	0	129	122	143	337	203	0
7	85	290	240	0	0	204	302	39	245	150
8	34	72	336	280	194	218	40	345	52	153
9	0	0	0	0	0	0	0	0	0	0
10	28	327	82	157	50	230	83	138	200	233
11	177	26	44	0	29	27	263	137	287	244
12	320	202	350	292	0	0	142	18	187	0
13	255	226	346	0	33	207	243	114	220	125
14	296	340	0	0	163	166	89	120	299	338
15	141	241	235	247	0	272	276	19	78	274
16	64	102	5	2	168	49	17	11	339	278
17	108	158	174	214	43	119	322	130	140	162
18	229	0	0	0	0	0	0	0	0	0
19	112	184	275	73	172	0	262	101	66	267
20	285	149	148	208	65	48	227	46	268	0
21	185	22	277	210	215	223	251	261	93	0
22	317	13	197	0	84	115	258	0	0	0
23	273	7	87	59	70	232	110	107	297	213
24	264	0	105	0	0	0	10	199	0	0
25	31	179	96	0	332	0	23	170	144	160
26	286	269	316	118	134	250	331	314	186	289
27	0	0	0	0	0	0	192	0	0	0
28	321	57	6	319	257	181	75	246	318	310
29	99	305	86	146	167	173	182	348	301	55
30	32	68	333	0	0	0	136	69	0	98
31	183	35	0	0	0	127	254	224	313	0
32	315	266	0	222	67	25	238	156	219	133
33	94	0	328	0	0	109	284	3	212	45
34	145	324	61	249	0	237	76	175	306	159
35	198	308	123	1	117	164	252	325	298	311
36	0	0	0	0	0	0	0	0	0	0
37	193	24	231	201	14	103	171	259	211	209
38	42	37	303	0	56	236	191	295	293	95
39	88	81	341	0	309	106	294	80	347	0
40	90	62	228	104	21	151	126	152	342	97
41	256	41	242	291	0	71	180	176	53	281
42	147	132	344	195	124	58	121	205	326	116
43	225	300	178	79	16	221	265	217	155	60
44	12	253	161	288	196	139	260	51	349	92
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 20: Melhor solução encontrada para a Instância No. 15.

Função objetivo: $v = 40$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	241	331	43	160	316	272	252	169	289	327
2	39	13	187	147	130	88	0	177	11	323
3	319	69	0	0	150	198	0	97	155	254
4	204	182	340	158	347	210	21	133	75	277
5	105	101	328	94	282	0	7	12	42	71
6	124	188	0	0	215	224	138	219	137	332
7	300	131	121	0	34	20	0	246	291	232
8	333	259	48	310	73	108	264	125	235	89
9	0	0	0	0	0	0	0	0	0	0
10	255	77	18	31	197	163	195	154	146	57
11	134	55	269	0	226	35	267	123	203	15
12	0	44	189	0	99	0	168	53	296	180
13	179	279	4	0	95	0	253	111	156	242
14	202	70	217	0	56	128	102	0	24	49
15	41	19	50	0	270	152	234	0	233	200
16	85	82	337	0	321	78	157	90	223	257
17	237	271	213	0	109	81	64	165	258	28
18	0	0	0	0	0	0	0	0	0	0
19	285	275	322	0	261	52	129	281	306	104
20	54	286	185	274	314	61	0	32	58	103
21	148	305	229	287	220	308	22	86	149	262
22	63	307	336	212	37	46	208	186	110	139
23	170	74	0	0	159	172	342	51	344	218
24	276	142	0	0	132	0	324	91	140	216
25	190	2	330	67	0	72	230	348	184	183
26	334	265	23	343	317	60	171	302	194	100
27	0	0	0	0	0	0	0	0	0	0
28	96	176	206	225	346	66	301	250	278	87
29	79	40	247	0	162	145	30	290	221	205
30	5	263	236	0	0	107	283	0	122	0
31	164	62	298	0	284	36	6	211	33	93
32	127	0	181	175	0	26	339	345	248	27
33	114	256	299	0	0	313	0	280	214	326
34	207	65	245	315	293	178	0	201	0	9
35	294	228	251	297	318	304	196	329	106	98
36	0	0	0	0	0	0	0	0	0	0
37	17	47	126	119	288	115	68	76	311	238
38	209	153	222	80	191	320	0	117	118	84
39	173	141	25	0	0	161	143	341	136	174
40	151	16	112	0	227	231	338	273	335	199
41	113	349	192	0	303	14	116	325	38	243
42	92	120	167	8	295	0	350	249	3	292
43	268	240	166	0	0	10	144	59	244	1
44	29	135	309	239	266	45	260	312	193	83
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 21: Melhor solução encontrada para a Instância No. 16.

Função objetivo: $v = 21$

Hora	Salas de aula										
	1	2	3	4	5	6	7	8	9	10	11
1	352	208	81	152	203	428	33	137	98	109	93
2	83	31	336	46	61	202	298	260	23	339	369
3	255	258	364	271	84	117	60	13	319	365	155
4	102	383	74	417	206	389	437	400	368	326	162
5	27	153	3	288	239	283	72	374	134	115	305
6	390	268	281	276	397	252	324	49	355	418	183
7	2	373	406	230	204	399	315	7	52	35	244
8	224	122	157	287	132	344	168	89	227	114	55
9	0	0	348	0	0	0	0	0	0	0	0
10	107	329	97	111	21	280	12	402	225	11	5
11	301	332	212	6	286	277	112	19	32	257	94
12	158	292	82	166	363	116	78	9	403	229	311
13	62	34	274	297	420	95	106	0	331	148	291
14	50	325	350	38	302	349	213	194	313	337	375
15	57	247	149	290	144	189	439	371	285	0	150
16	347	392	65	241	195	76	362	267	42	125	36
17	249	235	151	404	342	250	196	47	282	407	4
18	0	0	0	0	0	0	0	0	0	0	0
19	312	88	104	14	59	256	70	259	440	296	165
20	232	8	199	105	427	433	41	401	295	37	234
21	253	121	211	90	138	25	436	68	26	379	188
22	56	410	143	238	266	66	172	269	413	30	431
23	221	1	22	240	40	353	320	136	423	351	300
24	113	0	408	273	190	388	409	193	127	176	306
25	387	358	377	237	422	424	435	184	24	180	275
26	45	415	16	376	167	17	20	18	223	191	29
27	0	0	0	0	0	0	0	0	0	0	130
28	378	87	382	317	161	333	412	222	92	343	118
29	236	15	215	248	278	357	243	10	303	411	110
30	328	164	54	154	289	185	75	318	366	163	293
31	381	179	205	85	421	340	133	28	145	304	86
32	129	299	251	430	131	201	91	393	99	341	187
33	438	43	254	140	123	160	218	103	262	139	231
34	174	220	226	346	156	200	370	181	263	120	270
35	51	119	396	192	80	39	359	233	323	198	216
36	0	0	0	0	0	0	0	0	0	0	142
37	284	228	159	419	186	432	44	367	135	146	53
38	386	175	338	69	425	101	141	261	294	316	219
39	209	177	321	124	426	434	310	416	173	414	309
40	73	210	178	217	345	429	64	279	265	356	108
41	246	335	391	380	242	398	77	147	171	245	314
42	327	100	207	170	96	322	63	67	354	360	307
43	48	71	395	405	182	330	308	372	384	264	214
44	394	79	197	385	361	58	169	272	126	334	128
45	0	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 22: Melhor solução encontrada para a Instância No. 17.

Função objetivo: $v = 119$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	42	244	104	220	320	57	152	111	49	275
2	255	108	0	263	3	348	36	109	120	32
3	238	265	250	196	55	349	0	52	184	86
4	295	235	258	60	201	0	338	178	180	150
5	0	217	208	195	187	0	63	225	151	20
6	71	268	345	149	9	6	0	280	204	21
7	97	192	58	252	248	309	283	293	70	267
8	81	47	288	313	74	41	35	279	37	141
9	0	0	0	0	0	0	0	0	0	0
10	88	17	270	259	19	317	318	0	85	7
11	87	84	247	72	147	0	175	45	240	0
12	339	158	186	164	121	0	0	100	306	177
13	83	215	0	145	94	341	233	310	325	22
14	67	239	307	299	296	179	0	274	329	133
15	1	277	289	0	0	340	254	28	262	93
16	210	102	227	82	165	251	0	326	193	257
17	170	75	221	301	46	213	0	69	292	114
18	0	0	0	0	0	0	0	0	0	0
19	202	128	2	249	112	237	77	223	76	214
20	185	324	350	0	163	0	4	0	54	269
21	129	172	260	33	298	0	0	304	271	284
22	134	98	332	0	228	0	305	31	39	337
23	327	24	12	243	242	0	18	169	13	168
24	253	142	30	0	79	273	0	0	34	43
25	167	95	231	73	0	0	278	194	346	101
26	15	137	119	291	130	333	154	124	115	335
27	0	0	0	0	0	0	0	0	0	0
28	160	344	171	0	132	48	182	110	286	44
29	148	166	330	189	342	68	316	328	266	224
30	139	256	99	0	308	0	0	0	26	282
31	51	53	96	117	209	206	0	91	66	319
32	281	8	103	65	276	11	80	297	246	334
33	118	144	0	14	212	0	205	64	229	0
34	287	264	123	140	59	56	0	198	302	105
35	27	89	23	234	336	183	0	127	162	290
36	161	0	0	0	0	0	0	0	0	0
37	29	113	261	181	116	218	0	126	61	146
38	241	78	138	0	323	199	16	203	322	135
39	62	176	226	0	0	0	10	321	136	294
40	90	230	106	159	50	343	245	232	222	211
41	200	125	303	300	285	347	0	131	0	25
42	207	315	312	40	314	272	0	153	331	92
43	143	157	216	0	311	5	188	0	38	190
44	155	219	174	122	156	197	173	107	191	236
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 23: Melhor solução encontrada para a Instância No. 18.Função objetivo: $v = 16$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	123	280	304	5	287	210	142	352	200	37
2	196	183	273	161	16	32	188	112	28	18
3	301	19	162	47	202	300	220	155	2	175
4	120	334	346	357	35	229	0	106	310	13
5	89	272	7	389	90	174	45	154	1	41
6	104	65	219	390	80	248	34	257	298	354
7	63	75	307	387	367	291	146	226	216	48
8	277	186	256	244	176	29	93	351	292	62
9	0	0	0	0	0	0	0	113	0	0
10	212	270	377	38	321	166	260	388	143	25
11	96	187	282	236	221	165	87	374	153	60
12	61	296	144	339	288	276	283	17	46	64
13	137	360	299	109	208	125	259	199	55	338
14	269	24	4	180	156	326	59	262	261	324
15	178	358	140	251	240	295	241	194	107	320
16	384	218	110	203	281	53	344	179	255	232
17	101	69	195	205	303	213	56	353	102	204
18	0	0	0	268	0	0	0	0	0	0
19	371	333	51	372	172	284	264	168	263	317
20	85	249	15	368	33	207	343	366	39	197
21	306	318	223	139	325	66	11	365	173	83
22	42	233	158	311	275	335	332	362	118	111
23	392	246	363	265	0	230	329	394	169	68
24	382	77	376	22	91	147	21	239	58	290
25	217	92	359	375	350	134	293	20	79	57
26	36	225	193	99	151	71	12	348	177	254
27	0	0	0	349	0	0	0	0	0	0
28	252	160	198	150	250	119	81	345	235	114
29	82	297	189	117	192	163	54	313	399	126
30	6	10	130	397	43	26	337	86	27	289
31	340	185	136	373	95	170	342	379	400	227
32	138	108	9	330	149	0	167	258	8	364
33	322	129	274	52	72	331	122	314	398	319
34	97	355	266	356	103	131	302	378	182	214
35	234	228	242	341	3	184	124	385	135	271
36	0	0	0	0	0	132	0	0	0	0
37	386	14	157	49	361	285	31	145	128	316
38	206	327	267	243	78	315	237	74	279	164
39	105	201	70	211	116	23	98	395	67	245
40	215	336	347	133	190	247	84	370	253	148
41	312	309	121	231	224	40	191	383	127	328
42	391	222	94	181	152	0	171	396	44	159
43	294	369	30	50	238	76	100	286	381	88
44	115	73	278	393	308	305	209	141	380	323
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 24: Melhor solução encontrada para a Instância No. 19.

Função objetivo: $v = 97$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	157	309	193	163	319	303	153	103	152	369
2	201	346	132	45	165	231	142	175	60	385
3	202	208	190	143	289	239	232	248	378	84
4	225	30	280	158	53	124	317	99	381	243
5	177	238	51	68	286	156	15	348	352	167
6	123	365	269	146	92	244	29	2	20	251
7	210	8	113	187	268	141	380	59	192	250
8	313	377	273	154	310	267	48	264	57	41
9	0	0	0	0	0	0	0	0	0	0
10	28	213	282	397	171	252	333	7	39	71
11	316	258	13	347	364	253	98	395	72	263
12	266	373	33	298	275	276	24	314	172	69
13	366	191	249	338	18	110	265	228	284	161
14	145	356	396	129	274	205	335	376	353	390
15	337	151	306	358	270	293	318	259	325	229
16	235	218	221	342	288	23	247	360	386	199
17	102	370	86	149	107	237	343	100	55	150
18	0	0	0	0	0	0	0	0	301	388
19	227	357	131	399	67	139	27	3	272	168
20	46	291	304	117	108	197	222	359	240	65
21	219	50	97	36	1	77	255	125	115	82
22	196	340	21	189	111	169	188	368	328	371
23	198	324	256	183	279	299	62	355	6	362
24	181	384	52	138	166	0	297	133	43	204
25	106	375	5	245	321	179	217	184	87	40
26	374	147	140	211	95	332	70	322	101	83
27	0	0	0	0	0	0	0	0	0	0
28	42	182	331	94	173	277	176	206	323	262
29	49	79	155	134	327	295	122	64	104	278
30	285	81	160	12	300	271	85	90	341	315
31	230	261	363	164	35	130	80	287	14	25
32	185	391	9	329	326	120	178	308	216	292
33	207	379	78	194	281	0	137	349	361	0
34	383	290	31	16	294	283	226	215	75	351
35	382	63	93	10	58	37	392	121	4	350
36	0	0	0	0	0	0	0	0	241	0
37	96	296	127	200	246	170	34	26	105	128
38	220	76	209	339	174	344	144	116	56	186
39	109	307	398	302	126	17	148	47	19	389
40	38	44	394	223	118	73	159	180	234	387
41	242	11	254	214	311	135	61	367	400	89
42	312	336	114	393	260	236	91	136	305	257
43	88	119	320	345	22	74	330	54	354	162
44	66	233	112	224	195	212	334	372	32	203
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Tabela 25: Melhor solução encontrada para a Instância No. 20.

Função objetivo: $v = 13$

Hora	Salas de aula									
	1	2	3	4	5	6	7	8	9	10
1	198	28	255	2	179	162	327	0	89	158
2	189	0	114	288	160	328	0	319	72	315
3	329	0	295	281	286	337	137	83	171	205
4	296	282	6	56	39	0	272	97	77	0
5	188	111	237	65	247	0	172	0	290	110
6	0	0	64	0	348	124	59	0	234	98
7	14	223	85	0	0	17	96	276	214	67
8	156	0	190	269	250	11	125	21	233	349
9	0	0	0	0	0	0	0	0	0	0
10	133	341	109	5	99	344	35	0	245	229
11	324	336	210	0	192	61	146	127	170	238
12	0	200	239	307	154	34	23	244	0	311
13	285	27	230	215	275	321	184	249	139	258
14	217	226	191	155	44	144	195	136	256	287
15	0	0	347	0	1	166	7	24	112	326
16	73	177	298	251	100	279	107	118	101	161
17	32	0	131	50	150	119	66	243	303	271
18	0	0	0	0	0	0	0	0	0	0
19	113	0	38	219	331	10	79	241	218	88
20	340	69	235	22	30	317	204	63	57	313
21	0	0	102	108	70	199	0	299	134	345
22	60	342	142	53	0	13	40	260	211	104
23	90	350	26	46	148	159	9	0	314	147
24	0	216	182	305	93	115	84	135	312	0
25	58	343	157	12	37	333	86	266	145	8
26	87	0	277	68	254	132	41	0	300	33
27	0	0	48	0	0	0	0	0	0	0
28	80	75	186	268	270	208	117	236	31	289
29	167	231	212	92	325	338	81	222	213	301
30	278	95	126	201	0	261	152	0	25	82
31	105	54	203	322	330	165	181	151	320	121
32	339	0	221	140	0	175	4	252	202	183
33	227	169	242	253	332	263	176	0	302	346
34	280	51	259	163	18	15	0	225	207	335
35	120	123	262	185	138	16	45	71	274	173
36	0	0	0	0	0	0	0	0	0	0
37	129	228	240	310	43	62	178	143	187	316
38	49	42	224	267	164	128	141	284	291	168
39	0	304	334	248	292	106	0	0	0	153
40	19	323	52	306	220	91	29	103	297	20
41	318	209	55	36	194	273	283	122	116	0
42	0	0	94	196	180	308	0	78	0	0
43	74	294	3	293	193	264	76	0	0	174
44	309	206	232	265	257	149	47	197	246	130
45	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor.

Capítulo 7

Conclusões

Neste trabalho foi desenvolvido um algoritmo VNS especializado aplicado ao problema de otimização de grade horária e para o tipo de problema idealizado pela Competição Internacional de Grade Horária. Foram realizados testes com as 20 instâncias disponíveis e foram verificados que essas instâncias são casos típicos de problemas de grande complexidade e raramente uma técnica de otimização encontra a solução ótima desse tipo de problema para as instâncias disponíveis.

O algoritmo VNS desenvolvido se mostrou eficiente para encontrar soluções de excelente qualidade das 20 instâncias do problema de grade horária. Assim, o desempenho do algoritmo VNS se mostrou como sendo um dos mais eficientes para resolver esse tipo de problema e para essas instâncias complexas. Deve-se observar que no problema de otimização de grade horária não são eficientes as propostas tradicionais de especificar um tipo de vizinhança e de avaliar a qualidade e a infactibilidade de cada tipo de vizinhança de forma independente. Essa dificuldade acontece porque mesmo nas estruturas de vizinhanças muito simples o número de propostas de solução vizinhas disponíveis para avaliação são muito grandes e, portanto, precisa-se de um tempo de processamento muito elevado para identificar uma proposta de solução vizinha. Adicionalmente, como acontece com muitos problemas de otimização de sistemas de energia elétrica, não é simples implementar uma técnica de redução de vizinhança que seja simples e eficiente de implementar.

Deve-se mencionar que nas fases iniciais deste trabalho de pesquisa foram idealizados e implementados vários tipos de algoritmos VNS em que as estruturas de vizinhança foram idealizadas e implementadas na forma tradicional. Assim, as estruturas de vizinhança implementadas devido ao esforço de processamento foram muito mais simples que as idealizadas e implementadas neste trabalho. Dessa forma, nesses algoritmos VNS implementados não foram verificados desempenhos competitivos. Assim, por exemplo, para as quatro instâncias do problema de otimização de grade horária, as melhores soluções encontradas oscilaram entre valores da função objetivo na faixa de 150 a 300.

A proposta apresentada neste trabalho, a formulação de estruturas de vizinhança e principalmente na forma de encontrar a melhor solução vizinha resolvendo um problema de menor

tamanho e muito mais rápido de resolver comparado com o processo de resolução do problema completo, mostrou-se muito eficiente. Essa proposta de otimização é inovadora e pode ser facilmente estendida, especializada e aplicada na resolução de vários problemas de otimização de sistemas de energia elétrica onde aparecem o mesmo tipo de problema que no problema de otimização de grade horária, isto é, a formulação de estruturas de vizinhança de cardinalidade muito elevada (uma vizinhança com muitas propostas de solução vizinhas) e com muitas propostas de solução vizinhas de pobre qualidade.

O modelo matemático exato e completo apresentado neste trabalho foi idealizado e desenvolvido no LAPSEE pelo Professor John Fredy Franco Baquero. Entretanto, a importância desse modelo não era considerado relevante porque quando esse modelo era usado para resolver algumas instâncias da Competição Internacional de Grade Horária, o processo terminava após estourar a capacidade de memória para armazenamento da informação, apresentando uma solução de qualidade mediana e com informação de qualidade pobre em relação com o *gap*. Entretanto, esse modelo se tornou fundamental para desenvolver o algoritmo VNS apresentado neste trabalho.

Em resumo, o algoritmo VNS desenvolvido neste trabalho se mostrou muito competitivo e inovador em algumas fases da implementação. Assim, em trabalhos futuros podem ser desenvolvidos os seguintes trabalhos de pesquisa:

- Melhorar e ordenar de forma mais adequada as estruturas de vizinhança do problema de otimização de grade horária para encontrar soluções de melhor qualidade que os encontrados com a versão corrente do algoritmo VNS.
- Reformular e melhorar a proposta de vizinhança e encontrar a melhor solução vizinha usando a lógica da heurística de partição para melhorar o desempenho da proposta integral.
- Estender as propostas desenvolvidas neste trabalho na otimização de problemas de sistemas de energia elétrica como os abordados em ROMERO et al. (2007).

REFERÊNCIAS

ABRAMSON, D. Constructing school timetables using simulated annealing: sequential and parallel algorithms. **Management Science**, Catonsville, v. 37, p. 98-113, 1991.

ABRAMSON, D.; ABELA, J. A parallel genetic algorithm for solving the school timetabling-problem. In: AUSTRALIAN COMPUTER SCIENCE CONFERENCE, 15 th., 1992, [S.I.]. **Proceedings** [...][S.I.: s.n.], 1992.

AKKAN, C., GULCU, A. A bicriteria hybrid genetics algorithm with robustness objective for the course timetabling problem. **Computers and Operations Research**, Kidlington, v. 90, p. 22-32, 2018.

ALVAREZ-VALDÉS, R.; CRESPO, E.; TAMARIT, J.M. Design and implementation of a course scheduling system using tabu search. **European Journal of Operational Research**, Amsterdam, v. 137, n. 3, p. 512-523, 2002.

ALVAREZ-VALDÉS, R.; MARTIN, G.; TAMARIT, M. Constructing good solutions for the spanish school timetabling problem. **Journal of the Operational Research Society**, London, v. 47, p. 1203-1215, 1996.

AL-YAKOOB, S.M.; SHERALI, H.D. A mixed integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations. **European Journal of Operational Research**, Amsterdam, v. 180, n. 3, p. 1028-1044, 2007.

AL-YAKOOB, S.M.; SHERALI, H.D. Mathematical programming models and algorithms for a class faculty assignment problem. **European Journal of Operational Research**, Amsterdam, v. 173, n. 2, p. 488-507, 2006.

AVELLA, P.; VASILÉV, I. A computational study of a cutting plane algorithm for university course timetabling. **Journal of Scheduling**, New York, v. 8, p. 497-514, 2005.

BABAEI, H., KARIMPOUR, J., HADIDI, A. A survey of approaches for university course timetabling problem. **Computers & Industrial Engineering**, New York, v. 86, p. 43-59, 2015.

BAGGER, N.C.F., SORENSEN, M., STIDSEN, T.R. Dantzig-Wolfe decomposition of the daily course pattern formulation formulation for curriculum-based course timetabling. **European Journal of Operational Research**, Amsterdam, v. 272, p. 430-446, 2019.

BIRBAS, T.; DASKALAKI, S.; HOUSOS, E. Timetabling for greek high schools. **Journal of the Operational Research Society**, London, v. 48, p. 1191-1200, 1997.

BRAILSFORD, S.C.; POTTS, C.N.; SMITH, B.M. Constraint satisfaction problems: Algorithms and applications. **European Journal of Operational Research**, Amsterdam, v. 119, n. 3, p.

557-581, 1999.

BURKE, E.; CARTER, M. The practice and theory of automated timetabling II. In: INTERNATIONAL CONFERENCE ON THE PRACTICE AND THEORY OF AUTOMATED TIMETABLING, 2 st, 1997, Toronto. **Proceedings** [...] Toronto: University of Toronto, 1998.

BURKE, E.; ERBEN, W. The practice and theory of automated timetabling III. In: INTERNATIONAL CONFERENCE ON THE PRACTICE AND THEORY OF AUTOMATED TIMETABLING, 3 st, 2000, Konstanz. **Proceedings** [...] Heidelberg, 2000.

BURKE, E.; ROSS, P. The practice and theory of automated timetabling. In: INTERNATIONAL CONFERENCE ON THE PRACTICE AND THEORY OF AUTOMATED TIMETABLING, 1 st, 1995, Napier. **Proceedings** [...] Heidelberg, 1996.

BURKE, E.K.; JACKSON, K.; KINGSTON, J.H., WEARE, R. Automated timetabling: The state of the art. **The Computer Journal**, Oxford, v. 40, n. 9, p. 565-571, 1997.

BURKE, E.K., MCCOLLUM, B., MEISELS, A.; PETROVIC, S., QU, R. A graph based hyper-heuristic for educational timetabling problems. **European Journal of Operational Research**, Amsterdam, v. 176, n. 1, p. 177-192, 2007.

BURKE, E.K., ECKERSLEY, A.J., MCCOLLUM, B., PETROVIC, S., QU, R. Hybrid variable neighborhood approaches to university exam timetabling. **European Journal of Operational Research**, Amsterdam, v. 206, n. 1, p. 46-53, 2010.

BURKE, E.K.; PETROVIC, S. Recent research directions in automated timetabling. **European Journal of Operational Research**, Amsterdam, v. 140, n. 2, p. 266-280, 2002.

BURKE, E.K.; PETROVIC, S. Timetabling and rostering. **European Journal of Operational Research**, Amsterdam, v. 153, n. 1, p. 1-12, 2004.

CARTER, M.W.; LAPORTE, G. Recent developments in practical examination timetabling. In: CARTER, M.W. **A comprehensive course timetabling and student scheduling system at the University of Waterloo**. Berlin: Springer-Verlag, 2001, p. 3-21 (Lecture Notes in Computer Science, 2079).

CHU, P.; BEASLEY, J.E. A genetic algorithm for the generalized assignment problem. **Computers and Operations Research**, Kidlington, v. 24, n. 1, p. 17-23, 1997.

COLORNI, A.; DORIGO, M.; MANIEZZO, V. Metaheuristics for high-school timetabling. **Computational Optimization and Applications**, New York, v. 9, n. 3, p. 277-298, 1998.

COOPER, T.B.; KINGSTON, J.H. The complexity of timetabling construction problems. In: CARTER, M.W. **A comprehensive course timetabling and student scheduling system at the University of Waterloo**. Berlin: Springer-Verlag, 2001, p. 283-295 (Lecture Notes in Computer Science, 2079).

- COSTA, D. Tabu search algorithm for computing an operational timetable. **European Journal of Operational Research**, Amsterdam, v. 76, n. 1, p. 98-110, 1994.
- CUNG, V. Strategies for the parallel implementation of metaheuristics. In: *Essays and Surveys in Metaheuristics*. [S.l.]: Kluwer Academic Publishers, Dordrecht, 2002. p. 263-308.
- DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. **Operations Research**, Catonsville, v. 8, n. 1, p. 101-111, 1960.
- DASKALAKI, S. An integer programming formulation for a case study in university timetabling. **European Journal of Operational Research**, Amsterdam, v. 153, p. 117-135, 2004.
- DASKALAKI, S.; BIRBAS, T. An integer programming formulation for a case study in university timetabling. **European Journal of Operational Research**, Amsterdam, v. 153, n. 1, p. 117-135, 2004.
- DASKALAKI, S.; BIRBAS, T. Efficient solutions for a university timetabling problem through integer programming. **European Journal of Operational Research**, Amsterdam, v. 160, n. 1, p. 106-120, 2005.
- DE CAUSMAECKER, P.; DEMEESTER, P.; VANDEN BERGHE, G. A decomposed metaheuristic approach for a real world university timetabling problem. **European Journal of Operational Research**, Amsterdam, v. 195, n. 1, p. 307-318, 2009.
- DIMOPOULOU, M.; MILLOTIS, P. Implementations of a university course and examination-timetabling system. **European Journal of Operational Research**, Amsterdam, v. 130, n. 1, p. 202-213, 2001.
- DREXL, A.; SALEWSKI, F. Distribution requirements and compactness constraints in school timetabling. **European Journal of Operational Research**, Amsterdam, v. 102, n. 1, p. 193-214, out. 1997.
- FERLAND, J.; LAVOUE, A. Exchange procedures for timetabling problems. **Discrete Applied Mathematics**, Amsterdam, p. 237-253, 1992.
- FONSECA, G.H.G., SANTOS, H.G., CARRANO, E.G. Integer programming techniques for educational timetabling. **European Journal of Operational Research**, Amsterdam, v. 262, p. 28-39, 2017.
- GANS, O. D. A computer timetabling system for secondary schools in the netherlands. **European Journal of Operational Research**, Amsterdam, v. 7, p. 175-182, 1981.
- GENDREAU, M.; HERTZ, A.; LAPORTE, G. A tabu search heuristic for the vehicle routing problem. **Management Science**, Catonsville, v. 40, n. 10, p. 1276-1290, out. 1994.

GLOVER, F. Future paths for integer programming and artificial intelligence. **Computers & Operations Research**, Kidlington, v. 13, p. 533-549, 1986.

GLOVER,F.; LAGUNA, M. Tabu search. **Kluwer Academic Publishers**, Dordrecht, 1997.

GLOVER,F.; KOCHENBERGER, G. Handbook of metaheuristics. **Kluwer Academic Publishers**, Dordrecht, 2003.

HANSEN, P. The steepest ascent mildest descent heuristic for combinatorial programming. In: CONGRESS ON NUMERICAL METHODS IN COMBINATORIAL OPTIMIZATION, 1986, Capri. **Proceedings** [...][S.I.: s.n.], 1986.

HANSEN, P.; MLADENOVIC, N. Variable neighborhood search: Principles and applications. **European Journal of Operations Research**, Amsterdam, v. 130, p. 449-467, 2001.

HANSEN, P.; MLADENOVIC, N. Variable neighborhood search. In: GLOVER, F.; KOCHENBERGER, G. (ed.) Handbook of metaheuristics, New York: Springer, 2003. p. 145-184

HERTZ, A. Tabu search for large scale timetabling problems. **European Journal of Operational Research**, Amsterdam, v. 54, n. 1, p. 39-47, 1991.

HERTZ, A.; WERRA, D. Using tabu search techniques for graph coloring. **Computing**, Wien, v. 39, n. 4, p. 345-351, 1987.

INTERNATIONAL TIMETABLING COMPETITION. available www.idsia.ch/Files/ttcomp2002.

KAHAR, M.N.M.; KENDALL, G. The examination timetabling problem at University Malaysia Pahang: Comparision of a constructive heuristic with an existing software solution. **European Journal of Operational Research**, Amsterdam, v. 207, n. 2, p. 557-565, 2010.

KANG, L.; WHITE, G.M. A logical approach to the resolution of constraints in timetabling. **European Journal of Operational Research**, Amsterdam, v. 61, n. 3, p. 306-317, 1997.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, Washington, v. 220, p. 671-680, 1983.

LAWRIE, N. An integer linear programming model of a school timetabling problem. **Computer Journal**, Oxford, v. 12, p. 307-316, 1969.

LU, Z.P.; HAO, J.K. Adaptive tabu search for course timetabling. **European Journal of Operational Research**, Amsterdam, v. 200, n. 1, p. 235-244, 2010.

MEISELS, A.; KAPLANSKY, E. Iterative restart technique for solving timetabling problems. **European Journal of Operational Research**, Amsterdam, v. 153, n. 1, p. 41-50, 2004.

- NONOBE, K.; IBARAKI, T. A tabu search approach to the constraints satisfaction problem as a general problem solver. **European Journal of Operational Research**, Amsterdam, v. 106, n. 2-3, p. 599-623, 1998.
- PAPOUTSIS, K.; VALOUXIS, C.; HOUSOS, E. A column generation approach for the timetabling problem of greek high schools,. **Journal of the Operational Research Society**, London, v. 54, p. 230-238, 2003.
- QU, R.; BURKE, E.K.; MCCOLLUM, B. Adaptive automated construction of hybrid heuristics for exam timetabling and graph coloring problems. **European Journal of Operational Research**, Amsterdam, v. 198, n. 2, p. 392-404, 2009.
- RESENDE, M. An introduction to GRASP, AT & T Lab, 458 páginas (material fornecido no XXXIX SBPO Simpósio Brasileiro de Pesquisa Operacional), Fortaleza, 2008.
- RESENDE, M.; RIBEIRO, C. Greedy randomized adaptive search procedures. In: **Handbook of Metaheuristics**, [S.l.]: Kluwer, Dordrecht, p. 219-249, 2003.
- ROMERO, R.; RIDER, M.J.; SILVA, I.J. A metaheuristic to solve the transmission expansion problem. **IEEE Transactions on Power Systems**, Piscataway, v. 22, n. 4, p. 2289-2291, 2007.
- ROMERO, R.; RIDER, LAVORATO, M.. Introdução a meta-heurísticas aplicadas a sistemas elétricos de potência, Minicurso do Simpósio Brasileiro de Sistemas Elétricos, SBSE, Goiania, 2012.
- SANTOS, H. G. Formulações e algoritmos para o problema de programação de horários em escolas. 2007. 142 f. Tese (Doutorado) Universidade Federal Fluminense, Niterói.
- SAVINIEC, L., SANTOS, M.O., COSTA, A.M. Parallel local search algorithms for high school timetabling problems. **European Journal of Operational Research**, Amsterdam, v. 265, p. 81-98, 2018.
- SCHAERF, A. A survey of automated timetabling. **Artificial Intelligence Review**, Dordrecht, v. 13, n. 2, p. 87-127, 1999.
- SCHAERF, A. Local search techniques for large high school timetabling problems. **IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans**, Piscataway, v. 29, n. 4, p. 368-377, 1999.
- SORIA-ALCARAZ, J.A., OZCAN, E., SWAN, J., KENDALL, G., CARPIO, M. Iterated local search using an add and delete hyper-heuristic for university course timetabling. **Applied Soft Computing**, Amsterdam, v. 40, p. 581-593, 2018.
- SOUZA, M. Programação de Horários em Escolas: Uma Aproximação por Meta-heurísticas. 2000. 167 f. Tese (Doutorado). Universidade Federal do Rio de Janeiro - COPPE/UFRJ, Rio de Janeiro.

SOUZA, M.; MACULAN, N. OCHI, L. Melhorando quadros de horários de escolas através de caminhos mínimos, **Tendências em Matemática Aplicada e Computacional**, Petropolis, v. 1, n. 1, p. 515-524, 2000.

SOUZA, M.; OCHI, L.; MACULAN, N. A GRASP-tabu search algorithm for solving school timetabling problems. In: RESENDE M.G.C., PINHO DE SOUZA, J. **Metaheuristics: Computer Decision-Making**. [S.l.]: Kluwer Academic Publishers, Dordrecht, 2003. p. 659-672.

STEFANO, C. D.; TETTAMANZI, A. An evolutionary algorithm for solving the school timetabling problem. In: AL., E. B. et (Ed.). **Proceedings of the EvoWorkshops on Applications of Evolutionary Computing**. London: Springer-Verlag, 2001, v. 2037, p. 452-462.

TORREÃO, J. R. A.; ROE, E. Microcanonical optimization applied to visual processing. **Physics Letters**, Amsterdam, v. 205, p. 377-382, 1995.

VANDEN BROEK, J.; HURKENS, C.; WOEGINGER, G. Timetabling problems at the TU Eindhoven. **European Journal of Operational Research**, Amsterdam, v. 196, n. 3, p. 877-885, 2009.

WERRA, D. An introduction to timetabling. **European Journal of Operational Research**, Amsterdam, v. 19, n. 2 , p. 151-162, 1985.

WERRA, D. The combinatorics of timetabling. **European Journal of Operational Research**, Amsterdam, v. 96, n. 3, p. 504-513, 1997.

ZHANG, D.F.; LIU, Y.K.; M'HALLAH, R.; LEUNG, S.C.H. A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problem. **European Journal of Operational Research**, Amsterdam, v. 203, n. 3, p. 550-558, 2010.

Apêndice A

Dados das Instâncias Usadas em Testes

Neste Apêndice são mostrados os dados de 4 das 20 instâncias da Competição Internacional de Otimização do Problema de Grade Horária. A formatação dos dados obedecem à estrutura de dados que são usados como dados de entrada do algoritmo VNS e, portanto, esses dados não se encontram tabulados da mesma forma que no site da Competição Internacional de Otimização do Problema de Grade Horária.

A interpretação dos dados é muito simples. Em cada linha aparece o evento, o número de salas em que pode ser programado o evento, a identificação das salas que podem ser usadas, o número de alunos matriculados no evento e a identificação dos alunos matriculados no evento. Assim, por exemplo, na Instância No. 1, o evento 2, pode ser programado em duas salas (as salas 1 e 5) e existem 9 alunos matriculados (os alunos 17, 99, 176, 177, 178, 179, 182, 183 e 186).

Tabela 26: Banco de dados da Instância No. 1.

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos alunos
1	1	3	9	86, 142, 143, 144, 145, 146, 147, 148, 199
2	2	1, 5	9	17, 99, 176, 177, 178, 179, 182, 183, 186
3	2	5, 9	11	98, 142, 144, 146, 148, 150, 152, 154, 157, 164, 188
4	2	4, 5	10	127, 129, 131, 133, 135, 137, 139, 141, 143, 145
5	2	5, 7	11	98, 130, 131, 132, 133, 134, 135, 136, 137, 138, 142
6	2	4, 5	8	40, 41, 42, 43, 44, 45, 60, 64
7	1	3	8	119, 120, 121, 122, 123, 124, 125, 133
8	2	5, 9	0	
9	2	3, 6	10	62, 63, 64, 65, 66, 67, 68, 69, 70, 71
10	1	5	11	5, 99, 167, 169, 172, 175, 178, 183, 188, 195, 200
11	1	6	11	6, 7, 8, 9, 10, 11, 12, 13, 15, 17, 99
12	3	2, 4, 5	10	85, 86, 87, 88, 89, 90, 91, 92, 93, 94
13	1	5	10	184, 185, 186, 187, 188, 189, 190, 193, 195, 200
14	2	3, 6	7	9, 10, 11, 12, 13, 14, 100
15	2	2, 5	6	2, 7, 169, 170, 193, 197
16	4	4, 5, 7, 10	10	87, 88, 89, 90, 91, 92, 93, 94, 95, 96
17	6	2, 3, 5, 6, 9, 10	10	24, 26, 28, 30, 32, 39, 42, 45, 48, 51
18	1	1	9	46, 47, 48, 52, 54, 56, 61, 65, 73
19	3	4, 5, 7	9	99, 100, 101, 102, 103, 104, 105, 109, 142
20	1	3	8	51, 52, 55, 60, 63, 66, 72, 75
21	2	4, 5	3	62, 69, 70
22	1	3	10	63, 64, 65, 66, 67, 68, 69, 70, 79, 81
23	2	3, 6	8	2, 4, 8, 10, 18, 35, 45, 200
24	2	5, 7	10	1, 2, 3, 4, 5, 14, 16, 18, 199, 200
25	3	5, 7, 8	4	120, 121, 122, 134
26	4	2, 3, 5, 6	10	67, 68, 69, 70, 71, 72, 73, 74, 75, 76
27	2	3, 6	10	35, 36, 37, 38, 43, 45, 47, 49, 51, 53
28	2	3, 6	8	109, 110, 111, 112, 113, 114, 117, 143
29	1	5	10	19, 143, 145, 147, 149, 151, 153, 155, 160, 165
30	1	5	9	79, 80, 81, 82, 83, 84, 85, 86, 87
31	2	4, 5	9	3, 4, 5, 6, 7, 8, 9, 13, 24
32	2	4, 5	10	136, 137, 138, 139, 140, 141, 142, 143, 144, 180
33	1	3	9	166, 167, 168, 169, 170, 171, 172, 173, 185
34	2	3, 6	8	67, 68, 69, 70, 71, 72, 73, 74
35	2	3, 6	10	1, 2, 3, 194, 195, 196, 197, 198, 199, 200
36	2	4, 5	10	12, 13, 14, 15, 16, 17, 18, 19, 20, 24
37	1	1	7	174, 175, 177, 179, 181, 183, 186
38	1	1	10	115, 116, 117, 118, 119, 120, 122, 124, 131, 142
39	1	5	10	117, 119, 121, 123, 125, 127, 129, 131, 133, 143
40	2	3, 6	2	135, 137
41	2	5, 10	10	124, 125, 126, 130, 132, 134, 142, 145, 148, 180
42	1	3	9	164, 166, 167, 168, 169, 170, 172, 175, 180
43	1	1	10	146, 147, 148, 149, 150, 151, 152, 153, 154, 187
44	2	4, 5	9	21, 23, 24, 27, 30, 33, 36, 39, 42
45	1	3	10	166, 167, 168, 169, 170, 171, 172, 173, 175, 181

Fonte: Elaborado pelo autor.

Tabela 26: Banco de dados da Instância No. 1 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos alunos
46	2	1, 5	10	118, 120, 122, 124, 126, 128, 130, 132, 134, 144
47	3	4, 5, 7	6	97, 102, 103, 104, 109, 142
48	1	1	8	158, 159, 160, 161, 166, 168, 170, 181
49	2	3, 6	10	80, 82, 86, 89, 92, 95, 98, 101, 120, 143
50	1	5	10	1, 147, 149, 151, 153, 155, 157, 159, 182, 187
51	4	2, 5, 8, 9	9	129, 131, 133, 135, 141, 144, 147, 150, 153
52	2	2, 5	10	101, 102, 103, 104, 105, 106, 107, 108, 110, 146
53	1	10	6	139, 144, 146, 148, 150, 182
54	1	10	10	140, 141, 142, 143, 144, 145, 146, 147, 148, 182
55	1	3	9	25, 27, 29, 31, 37, 40, 43, 46, 49
56	2	2, 5	10	163, 164, 165, 166, 167, 168, 169, 173, 175, 182
57	1	1	10	36, 37, 38, 39, 40, 41, 48, 50, 52, 54
58	2	5, 7	8	147, 148, 150, 152, 154, 156, 160, 183
59	2	5, 10	7	11, 14, 190, 191, 192, 193, 194
60	2	1, 5	10	5, 8, 183, 184, 185, 187, 189, 191, 193, 195
61	1	3	10	51, 52, 53, 54, 55, 56, 57, 58, 59, 60
62	2	5, 9	10	73, 74, 75, 76, 77, 78, 79, 80, 81, 82
63	3	2, 3, 5	10	103, 104, 105, 106, 107, 108, 109, 125, 132, 143
64	5	2, 4, 5, 8, 9	10	115, 116, 117, 118, 119, 120, 121, 122, 123, 144
65	1	5	10	7, 11, 25, 179, 180, 181, 185, 187, 189, 191
66	2	3, 6	8	21, 22, 27, 29, 31, 36, 41, 46
67	1	10	10	2, 169, 170, 171, 172, 173, 175, 178, 183, 196
68	2	5, 8	10	61, 62, 63, 64, 65, 66, 67, 68, 69, 72
69	2	5, 7	8	49, 52, 55, 58, 67, 77, 91, 144
70	1	5	9	53, 54, 55, 56, 57, 58, 67, 69, 72
71	2	4, 5	8	53, 54, 58, 61, 64, 67, 70, 76
72	3	5, 7, 8	10	24, 26, 28, 30, 32, 34, 36, 38, 40, 42
73	1	5	10	166, 167, 168, 169, 170, 171, 172, 173, 176, 183
74	2	2, 5	10	24, 25, 26, 27, 28, 29, 30, 31, 32, 33
75	2	5, 7	10	84, 85, 86, 87, 88, 89, 90, 91, 92, 93
76	1	5	10	48, 49, 50, 51, 52, 53, 54, 55, 56, 57
77	2	2, 5	10	104, 105, 106, 107, 108, 109, 110, 111, 112, 183
78	2	3, 6	6	114, 115, 118, 120, 121, 122
79	2	2, 5	9	105, 106, 107, 109, 112, 115, 118, 121, 124
80	3	1, 3, 6	10	107, 108, 109, 110, 111, 112, 119, 121, 123, 145
81	3	4, 5, 7	10	3, 4, 5, 9, 11, 13, 15, 17, 19, 21
82	1	3	9	81, 82, 84, 86, 88, 90, 92, 94, 98
83	2	5, 7	10	78, 79, 80, 81, 82, 83, 84, 85, 86, 87
84	2	4, 5	7	189, 190, 191, 193, 195, 197, 199
85	1	1	9	56, 57, 59, 62, 65, 68, 71, 74, 77
86	2	1, 5	10	163, 164, 165, 166, 167, 168, 171, 180, 183, 186
87	4	2, 5, 8, 9	9	33, 34, 35, 36, 38, 41, 44, 47, 50
88	1	5	10	39, 40, 41, 42, 44, 46, 48, 50, 52, 54
89	1	1	10	128, 130, 132, 134, 136, 138, 140, 142, 144, 147
90	2	4, 5	8	159, 160, 162, 163, 164, 165, 176, 184

Fonte: Elaborado pelo autor.

Tabela 26: Banco de dados da Instância No. 1 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos alunos
91	3	5, 7, 9	6	4, 5, 6, 7, 8, 11
92	4	2, 5, 8, 9	10	114, 115, 116, 117, 118, 125, 128, 131, 134, 145
93	1	5	10	75, 76, 77, 78, 79, 80, 81, 82, 83, 84
94	2	2, 5	10	179, 180, 181, 182, 183, 184, 185, 193, 196, 200
95	3	2, 5, 9	8	47, 49, 51, 52, 53, 54, 55, 56
96	3	5, 7, 8	10	116, 117, 118, 119, 120, 121, 122, 123, 149, 185
97	1	3	9	34, 186, 187, 188, 189, 190, 191, 192, 193
98	1	10	10	44, 45, 46, 47, 48, 49, 50, 51, 55, 59
99	4	2, 3, 5, 6	10	35, 170, 171, 172, 173, 174, 175, 176, 177, 180
100	2	5, 8	10	17, 18, 19, 20, 21, 22, 25, 28, 31, 45
101	2	5, 9	10	1, 2, 3, 4, 5, 6, 7, 198, 199, 200
102	4	4, 5, 7, 8	8	117, 119, 121, 123, 125, 128, 132, 145
103	3	5, 7, 8	9	108, 109, 110, 111, 112, 113, 114, 115, 118
104	3	4, 5, 7	7	64, 65, 66, 67, 71, 82, 85
105	1	5	10	4, 80, 81, 82, 83, 88, 95, 98, 101, 145
106	1	5	6	63, 68, 72, 80, 83, 86
107	2	3, 6	10	55, 56, 57, 58, 59, 60, 61, 62, 66, 72
108	2	4, 5	9	43, 44, 45, 49, 51, 53, 55, 57, 73
109	2	5, 9	10	6, 168, 169, 170, 171, 172, 173, 174, 175, 176
110	2	4, 5	10	6, 184, 185, 186, 187, 188, 189, 190, 194, 196
111	2	5, 8	10	28, 29, 30, 31, 32, 33, 34, 35, 36, 37
112	2	3, 6	9	38, 39, 40, 42, 44, 48, 50, 52, 54
113	1	10	10	8, 148, 149, 150, 151, 152, 153, 154, 155, 156
114	2	4, 5	10	67, 69, 73, 76, 79, 82, 85, 88, 91, 94
115	2	2, 5	10	6, 131, 132, 133, 134, 135, 137, 139, 148, 151
116	1	5	9	53, 54, 55, 56, 57, 59, 61, 69, 73
117	3	2, 3, 5	3	78, 96, 99
118	2	5, 8	4	103, 104, 106, 110
119	4	5, 7, 8, 10	8	7, 141, 142, 143, 144, 145, 150, 152
120	1	3	10	63, 64, 65, 69, 71, 73, 75, 77, 79, 81
121	2	5, 8	6	106, 107, 118, 121, 134, 145
122	1	5	5	24, 26, 28, 30, 32
123	1	3	5	21, 23, 24, 25, 26
124	2	4, 5	10	7, 127, 128, 129, 130, 136, 138, 146, 149, 152
125	2	5, 9	10	7, 166, 167, 168, 169, 170, 171, 172, 173, 176
126	2	5, 8	10	8, 157, 158, 159, 160, 161, 162, 163, 164, 177
127	3	2, 3, 5	10	31, 32, 33, 34, 35, 36, 37, 38, 39, 40
128	2	5, 7	10	2, 4, 8, 21, 26, 192, 194, 196, 198, 200
129	3	4, 5, 9	10	9, 122, 123, 124, 125, 126, 127, 128, 129, 132
130	2	2, 5	10	85, 86, 87, 91, 93, 95, 97, 113, 118, 146
131	4	2, 3, 5, 6	10	88, 89, 90, 91, 92, 93, 94, 95, 96, 97
132	1	1	9	10, 11, 12, 13, 14, 15, 16, 17, 18
133	2	2, 5	10	55, 56, 57, 58, 59, 60, 61, 62, 69, 73
134	1	5	10	27, 28, 29, 30, 31, 32, 33, 34, 35, 55
135	2	5, 9	10	68, 69, 70, 71, 80, 83, 86, 89, 92, 95

Fonte: Elaborado pelo autor.

Tabela 26: Banco de dados da Instância No. 1 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos alunos
136	1	5	10	7, 8, 9, 10, 15, 23, 26, 29, 32, 46
137	2	2, 5	9	8, 174, 175, 176, 177, 178, 180, 181, 183
138	1	5	10	8, 88, 89, 90, 92, 94, 96, 110, 114, 119
139	1	1	5	126, 127, 128, 129, 146
140	3	5, 7, 8	10	8, 130, 131, 132, 133, 134, 135, 136, 137, 138
141	2	2, 5	10	9, 127, 128, 129, 130, 131, 132, 133, 148, 151
142	4	4, 5, 7, 10	10	88, 89, 90, 91, 92, 93, 94, 95, 97, 100
143	1	5	8	2, 14, 25, 180, 184, 187, 190, 193
144	1	5	8	43, 114, 116, 119, 122, 131, 135, 139
145	3	2, 5, 9	10	9, 110, 111, 112, 113, 123, 126, 129, 132, 135
146	2	4, 5	6	19, 20, 28, 30, 32, 47
147	2	5, 8	9	9, 160, 161, 162, 163, 164, 165, 166, 168
148	1	3	10	9, 102, 103, 104, 105, 106, 107, 108, 109, 110
149	2	4, 5	10	9, 159, 160, 161, 162, 163, 164, 165, 174, 177
150	1	5	8	20, 21, 22, 23, 24, 25, 26, 29
151	1	3	10	89, 90, 91, 92, 93, 94, 95, 96, 97, 98
152	2	5, 9	10	9, 185, 186, 187, 188, 189, 190, 191, 194, 196
153	2	5, 7	10	10, 133, 134, 135, 136, 137, 138, 139, 140, 141
154	2	5, 9	10	89, 90, 91, 92, 93, 94, 95, 96, 97, 98
155	4	4, 5, 7, 9	6	15, 195, 196, 197, 198, 199
156	2	2, 5	10	10, 132, 133, 134, 135, 136, 137, 139, 142, 145
157	3	4, 5, 7	10	10, 177, 178, 179, 180, 181, 182, 183, 184, 185
158	1	1	10	28, 29, 30, 31, 32, 33, 34, 35, 36, 37
159	1	3	3	20, 158, 166
160	2	5, 7	8	44, 115, 117, 120, 123, 132, 136, 140
161	2	5, 7	10	1, 2, 3, 4, 5, 6, 47, 198, 199, 200
162	3	2, 3, 5	2	66, 77
163	1	3	10	20, 188, 189, 190, 191, 192, 193, 194, 195, 197
164	1	10	8	10, 153, 154, 155, 156, 157, 160, 163
165	1	5	10	59, 60, 61, 62, 63, 64, 65, 66, 70, 74
166	1	5	10	10, 113, 114, 115, 116, 117, 118, 120, 122, 124
167	2	5, 10	10	35, 36, 37, 38, 39, 40, 41, 42, 43, 44
168	2	4, 5	10	10, 155, 156, 157, 158, 159, 160, 161, 162, 163
169	2	2, 5	2	48, 152
170	1	5	10	11, 139, 140, 141, 142, 143, 144, 146, 149, 152
171	2	3, 6	10	83, 84, 85, 86, 87, 88, 89, 90, 91, 94
172	3	4, 5, 7	3	175, 180, 181
173	1	10	10	33, 34, 35, 36, 37, 38, 39, 40, 41, 45
174	2	2, 5	10	11, 119, 120, 121, 122, 124, 127, 130, 133, 137
175	1	3	10	1, 2, 3, 4, 5, 11, 197, 198, 199, 200
176	3	4, 5, 8	9	11, 102, 103, 104, 106, 107, 108, 109, 110
177	2	5, 9	8	11, 154, 155, 156, 157, 158, 161, 164
178	1	3	8	58, 60, 62, 70, 73, 76, 82, 85
179	1	10	10	11, 152, 153, 154, 155, 156, 157, 158, 161, 184
180	3	5, 7, 8	10	12, 135, 136, 137, 138, 139, 140, 143, 146, 150

Fonte: Elaborado pelo autor.

Tabela 26: Banco de dados da Instância No. 1 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos alunos
181	1	1	10	12, 124, 125, 126, 127, 128, 129, 130, 131, 142
182	2	5, 7	7	12, 164, 165, 169, 171, 173, 176
183	2	4, 5	10	13, 152, 153, 154, 155, 156, 157, 158, 161, 176
184	2	2, 5	10	18, 19, 20, 25, 28, 31, 34, 37, 40, 43
185	4	2, 5, 8, 9	8	14, 146, 149, 151, 153, 155, 159, 177
186	2	5, 9	9	46, 47, 48, 50, 52, 54, 56, 60, 74
187	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	16, 17, 18, 19, 20, 21, 22, 23, 24, 25
188	1	10	10	25, 27, 29, 31, 33, 35, 37, 39, 41, 43
189	1	5	4	12, 150, 151, 152
190	2	3, 6	10	22, 140, 141, 142, 143, 145, 147, 149, 151, 153
191	1	1	8	74, 75, 76, 77, 79, 81, 84, 95
192	2	4, 5	9	21, 148, 150, 152, 154, 156, 158, 179, 198
193	1	5	10	13, 187, 188, 189, 190, 191, 192, 193, 194, 197
194	4	4, 5, 7, 8	8	12, 100, 102, 110, 113, 116, 119, 125
195	3	4, 5, 7	9	12, 102, 103, 104, 105, 110, 112, 114, 116
196	2	5, 8	9	52, 53, 54, 55, 56, 57, 58, 59, 75
197	2	5, 9	10	12, 86, 87, 88, 89, 90, 91, 92, 107, 110
198	2	2, 5	10	2, 5, 8, 12, 185, 186, 190, 193, 196, 199
199	2	2, 5	9	80, 83, 85, 87, 89, 91, 93, 97, 99
200	2	3, 6	9	19, 162, 163, 164, 165, 167, 169, 171, 177
201	1	1	3	37, 39, 40
202	4	1, 3, 6, 10	7	27, 98, 99, 100, 101, 103, 111
203	1	5	6	21, 154, 161, 164, 167, 184
204	4	2, 5, 8, 9	5	173, 180, 182, 184, 186
205	3	2, 5, 9	6	168, 169, 178, 180, 182, 185
206	2	4, 5	8	22, 50, 53, 56, 59, 68, 85, 92
207	2	5, 7	10	87, 88, 89, 90, 91, 92, 93, 94, 95, 96
208	2	5, 8	10	61, 62, 63, 64, 65, 66, 67, 68, 70, 76
209	1	10	9	34, 35, 40, 43, 46, 64, 69, 74, 79
210	2	4, 5	10	71, 72, 74, 77, 80, 83, 86, 89, 92, 95
211	2	5, 8	9	13, 165, 166, 167, 168, 169, 170, 171, 177
212	1	10	10	64, 65, 66, 67, 68, 69, 70, 71, 72, 77
213	3	2, 5, 9	7	13, 125, 126, 127, 128, 129, 130
214	3	4, 5, 7	4	20, 186, 187, 188
215	2	2, 5	10	25, 26, 27, 32, 35, 38, 41, 52, 56, 60
216	5	1, 2, 3, 5, 6	10	83, 84, 85, 86, 87, 88, 89, 90, 92, 96
217	2	3, 6	6	14, 125, 128, 131, 134, 138
218	2	5, 8	10	28, 29, 30, 33, 36, 39, 42, 53, 57, 61
219	1	3	10	14, 166, 167, 168, 169, 170, 171, 172, 182, 184
220	2	5, 8	10	5, 9, 22, 28, 182, 183, 184, 186, 188, 190
221	2	5, 9	10	48, 153, 154, 155, 156, 157, 158, 159, 178, 181
222	2	2, 5,	10	29, 156, 157, 158, 159, 160, 161, 162, 163, 177
223	3	4, 5, 8	9	26, 138, 139, 140, 141, 146, 148, 150, 154
224	1	10	10	3, 6, 9, 22, 187, 188, 191, 194, 197, 200
225	3	2, 4, 5	7	3, 14, 184, 187, 190, 194, 198

Fonte: Elaborado pelo autor.

Tabela 26: Banco de dados da Instância No. 1 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos alunos
226	1	5	9	23, 70, 71, 72, 73, 74, 75, 76, 101
227	3	4, 5, 7	9	15, 122, 123, 124, 126, 129, 132, 135, 139
228	2	4, 5	10	14, 126, 127, 128, 129, 130, 131, 140, 143, 146
229	3	4, 5, 10	10	49, 95, 96, 97, 98, 99, 100, 101, 102, 111
230	1	5	10	16, 17, 18, 19, 20, 26, 28, 30, 32, 34
231	4	2, 5, 8, 9	8	1, 3, 5, 9, 33, 38, 43, 49
232	2	2, 5	9	24, 78, 79, 80, 81, 82, 83, 84, 111
233	3	4, 5, 8	10	14, 105, 106, 107, 108, 109, 110, 111, 112, 121
234	3	5, 7, 8	9	29, 30, 31, 32, 33, 34, 35, 36, 37
235	1	10	10	16, 186, 187, 188, 189, 190, 191, 192, 195, 197
236	4	2, 5, 8, 9	9	43, 44, 45, 47, 48, 49, 50, 51, 52
237	3	2, 3, 5	10	15, 115, 116, 118, 120, 122, 124, 126, 129, 135
238	4	5, 7, 8, 10	9	155, 156, 157, 158, 159, 160, 161, 162, 178
239	2	2, 5	10	22, 134, 135, 136, 137, 138, 139, 140, 141, 142
240	1	10	10	15, 115, 116, 117, 118, 119, 120, 121, 122, 125
241	2	2, 5	10	24, 25, 26, 27, 28, 29, 30, 31, 32, 33
242	2	2, 5	10	7, 9, 11, 13, 15, 22, 25, 28, 31, 46
243	2	3, 6	10	1, 2, 3, 4, 6, 15, 197, 198, 199, 200
244	2	4, 5	10	20, 110, 111, 112, 113, 114, 115, 116, 117, 119
245	2	2, 5	9	89, 90, 91, 92, 93, 96, 99, 102, 146
246	2	5, 9	7	4, 16, 185, 188, 191, 195, 199
247	3	4, 5, 8	10	13, 14, 15, 16, 17, 18, 19, 20, 21, 23
248	1	1	9	1, 158, 159, 162, 164, 179, 184, 189, 196
249	2	5, 7	10	15, 191, 192, 193, 194, 195, 196, 197, 198, 199
250	2	3, 6	10	72, 73, 76, 78, 81, 84, 87, 90, 93, 96
251	1	3	9	2, 160, 161, 163, 165, 180, 185, 192, 197
252	3	2, 3, 5	5	24, 25, 26, 27, 28
253	2	4, 5	8	3, 168, 170, 173, 181, 186, 193, 198
254	3	4, 5, 8	10	25, 93, 94, 95, 96, 97, 98, 99, 100, 112
255	1	10	9	75, 76, 77, 78, 79, 80, 81, 82, 83
256	2	1, 10	10	36, 37, 41, 43, 45, 47, 49, 51, 53, 55
257	1	3	10	1, 3, 6, 10, 23, 30, 193, 195, 197, 199
258	2	3, 6	10	16, 133, 134, 135, 136, 137, 138, 139, 140, 141
259	2	4, 5	10	16, 143, 144, 145, 146, 147, 148, 149, 150, 151
260	1	5	10	17, 190, 191, 192, 193, 194, 195, 196, 197, 198
261	1	3	10	17, 106, 107, 108, 109, 111, 113, 115, 117, 126
262	2	5, 9	10	71, 72, 73, 74, 75, 76, 77, 78, 80, 82
263	2	4, 5	9	47, 124, 125, 126, 128, 129, 133, 137, 141
264	3	4, 5, 8	8	17, 158, 159, 160, 161, 162, 163, 178
265	3	5, 7, 10	9	23, 164, 165, 166, 167, 168, 176, 180, 199
266	2	5, 7	10	23, 24, 31, 34, 37, 40, 43, 54, 58, 83
267	2	2, 5	9	17, 97, 98, 99, 100, 101, 102, 103, 107
268	2	3, 6	9	1, 10, 15, 27, 196, 197, 198, 199, 200
269	1	5	8	67, 68, 69, 71, 74, 80, 83, 86
270	4	2, 3, 5, 6	9	11, 16, 28, 182, 183, 185, 188, 191, 194

Fonte: Elaborado pelo autor.

Tabela 26: Banco de dados da Instância No. 1 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos alunos
271	1	3	9	35, 188, 189, 190, 191, 192, 193, 194, 195
272	2	4, 5	10	37, 132, 133, 134, 135, 136, 137, 138, 139, 140
273	3	5, 7, 8	10	1, 2, 3, 4, 5, 18, 197, 198, 199, 200
274	2	1, 10	10	66, 67, 68, 70, 72, 74, 76, 78, 80, 82
275	2	5, 10	7	38, 176, 179, 181, 183, 185, 187
276	2	3, 6	10	26, 87, 88, 89, 91, 93, 95, 97, 99, 101
277	2	5, 9	10	8, 10, 12, 14, 16, 23, 26, 29, 32, 48
278	2	5, 8	3	16, 22, 200
279	2	5, 9	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 18
280	1	10	8	26, 170, 172, 174, 175, 176, 177, 178
281	2	4, 5	10	23, 123, 124, 125, 126, 127, 128, 129, 130, 135
282	2	2, 5	10	36, 37, 38, 39, 40, 41, 42, 43, 44, 50
283	1	3	10	1, 4, 7, 10, 24, 184, 189, 192, 195, 198
284	2	3, 6	10	52, 54, 56, 58, 60, 62, 65, 70, 75, 100
285	2	5, 9	10	10, 11, 12, 13, 14, 15, 16, 17, 18, 21
286	2	3, 6	8	18, 165, 166, 167, 168, 171, 174, 178
287	2	5, 8	9	62, 63, 64, 65, 66, 67, 68, 70, 77
288	2	5, 8	10	72, 73, 74, 75, 76, 77, 78, 79, 80, 81
289	2	5, 8	10	15, 16, 17, 26, 29, 32, 35, 38, 41, 44
290	3	2, 3, 5	10	49, 50, 51, 52, 53, 54, 55, 56, 57, 58
291	3	4, 5, 8	10	22, 101, 102, 103, 104, 105, 106, 107, 108, 120
292	2	3, 6	10	18, 189, 190, 191, 192, 193, 194, 195, 196, 197
293	2	5, 9	4	31, 102, 110, 112
294	3	2, 3, 5	10	19, 98, 99, 100, 104, 106, 108, 111, 115, 120
295	1	1	10	19, 127, 128, 129, 131, 133, 141, 144, 147, 151
296	1	1	9	12, 13, 14, 16, 24, 27, 30, 33, 51
297	1	3	9	47, 48, 49, 50, 52, 60, 62, 70, 78
298	3	5, 7, 9	8	19, 36, 37, 41, 44, 66, 71, 76
299	1	10	10	21, 177, 178, 179, 180, 181, 182, 183, 184, 185
300	2	3, 6	9	20, 189, 190, 191, 192, 193, 194, 195, 196
301	1	10	10	30, 99, 100, 101, 102, 103, 104, 105, 107, 111
302	3	4, 5, 8	7	5, 20, 186, 189, 192, 196, 200
303	2	5, 8	10	21, 126, 127, 128, 129, 130, 131, 132, 134, 147
304	3	4, 5, 8	4	22, 127, 130, 138
305	2	5, 9	8	22, 153, 155, 156, 157, 158, 159, 162
306	2	5, 9	10	13, 14, 15, 16, 17, 18, 19, 20, 21, 27
307	2	4, 5	3	22, 152, 155
308	1	3	10	23, 159, 160, 161, 162, 163, 164, 165, 174, 186
309	2	2, 5	10	6, 7, 8, 10, 12, 14, 16, 18, 20, 23
310	2	3, 6	8	74, 76, 77, 78, 79, 80, 81, 82
311	2	2, 5	4	23, 73, 87, 98
312	2	3, 6	10	49, 50, 51, 52, 53, 54, 55, 56, 57, 58
313	2	5, 9	10	25, 148, 149, 150, 151, 152, 153, 154, 155, 156
314	3	4, 5, 8	10	39, 144, 145, 146, 147, 148, 149, 150, 151, 152
315	2	5, 9	10	28, 29, 30, 31, 32, 33, 34, 35, 36, 38

Fonte: Elaborado pelo autor.

Tabela 26: Banco de dados da Instância No. 1 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos alunos
316	2	5, 9	8	61, 62, 63, 64, 65, 66, 71, 78
317	2	5, 9	10	88, 89, 90, 91, 92, 93, 94, 95, 96, 97
318	3	1, 3, 6	10	23, 126, 127, 128, 129, 130, 131, 132, 133, 153
319	2	3, 6	10	26, 101, 102, 103, 105, 107, 109, 112, 116, 121
320	2	5, 9	10	34, 35, 36, 37, 38, 39, 40, 41, 42, 43
321	1	10	9	11, 12, 13, 14, 15, 17, 34, 44, 50
322	4	2, 3, 5, 6	10	27, 151, 152, 153, 157, 159, 161, 163, 165, 178
323	1	5	10	7, 84, 181, 182, 186, 188, 190, 192, 194, 196
324	2	5, 9	9	1, 2, 3, 4, 5, 6, 7, 8, 200
325	2	3, 6	10	50, 149, 150, 151, 152, 153, 154, 155, 156, 157
326	4	1, 2, 3, 5	1	42
327	1	10	7	34, 162, 165, 166, 167, 174, 178
328	2	3, 6	9	121, 122, 123, 124, 125, 138, 141, 144, 147
329	2	2, 5	7	34, 159, 162, 165, 168, 174, 189
330	1	10	10	38, 157, 158, 159, 160, 161, 162, 178, 181, 184
331	3	2, 5, 9	10	39, 101, 102, 103, 104, 105, 106, 107, 108, 120
332	1	1	9	52, 53, 54, 55, 56, 57, 58, 59, 60
333	2	2, 5	10	39, 154, 155, 156, 158, 160, 162, 164, 174, 179
334	1	10	10	40, 101, 103, 108, 111, 114, 117, 120, 123, 178
335	2	3, 6	8	123, 124, 125, 126, 127, 128, 129, 139
336	1	10	10	40, 105, 106, 107, 108, 109, 110, 111, 112, 113
337	3	2, 5, 9	8	38, 39, 42, 45, 51, 67, 72, 77
338	2	2, 5	10	41, 84, 85, 87, 90, 93, 96, 99, 118, 139
339	3	2, 3, 5	4	192, 194, 196, 198
340	2	3, 6	5	25, 27, 29, 31, 33
341	2	5, 10	10	68, 70, 75, 78, 81, 84, 87, 90, 93, 96
342	2	5, 7	10	42, 43, 44, 45, 46, 47, 49, 51, 53, 56
343	2	2, 5	10	44, 112, 113, 114, 115, 116, 117, 118, 119, 155
344	2	4, 5	9	5, 7, 8, 9, 10, 11, 12, 13, 41
345	4	4, 5, 7, 10	8	140, 141, 142, 144, 145, 147, 150, 153
346	3	2, 5, 9	8	59, 60, 61, 62, 63, 64, 65, 79
347	1	5	9	85, 86, 90, 92, 94, 96, 98, 100, 112
348	2	5, 7	9	98, 181, 182, 183, 184, 185, 186, 187, 188
349	2	5, 7	8	12, 45, 179, 181, 186, 189, 192, 195
350	2	5, 9	10	19, 20, 21, 22, 23, 24, 25, 26, 27, 41
351	1	10	10	79, 100, 101, 102, 103, 104, 105, 106, 108, 112
352	2	2, 5	10	42, 100, 101, 102, 103, 104, 105, 106, 108, 113
353	2	5, 9	9	42, 160, 161, 162, 163, 174, 177, 181, 200
354	1	3	9	42, 130, 131, 132, 133, 134, 135, 136, 137
355	2	5, 9	9	49, 50, 51, 53, 55, 57, 59, 63, 81
356	2	4, 5	10	42, 104, 105, 106, 107, 108, 109, 110, 111, 113
357	2	4, 5	9	81, 83, 88, 91, 94, 97, 100, 119, 140
358	3	5, 9, 10	10	45, 110, 111, 112, 113, 114, 121, 123, 126, 140
359	3	5, 7, 8	9	73, 74, 75, 76, 77, 78, 79, 80, 81
360	2	4, 5	10	44, 96, 97, 98, 99, 101, 103, 105, 108, 113

Fonte: Elaborado pelo autor.

Tabela 26: Banco de dados da Instância No. 1 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos alunos
361	2	5, 9	7	44, 113, 115, 116, 117, 118, 140
362	1	1	9	45, 136, 137, 138, 139, 140, 149, 151, 179
363	1	1	8	45, 99, 100, 101, 102, 103, 104, 114
364	1	1	3	1, 6, 57
365	1	1	9	43, 44, 47, 49, 51, 53, 55, 57, 59
366	3	5, 7, 10	10	82, 172, 173, 174, 175, 176, 177, 179, 182, 185
367	1	10	10	58, 161, 162, 163, 164, 165, 166, 167, 179, 182
368	2	4, 5	8	59, 61, 72, 75, 78, 81, 84, 97
369	1	10	9	45, 46, 48, 50, 52, 54, 56, 58, 60
370	1	10	7	191, 192, 194, 196, 197, 198, 199
371	2	2, 5	9	130, 132, 134, 136, 142, 145, 148, 151, 154
372	2	4, 5	7	46, 113, 127, 130, 133, 136, 141
373	1	10	10	46, 137, 138, 139, 140, 143, 146, 149, 152, 156
374	1	5	10	1, 2, 3, 4, 5, 6, 7, 8, 82, 200
375	1	5	10	46, 113, 114, 115, 116, 117, 118, 119, 120, 143
376	1	10	9	60, 61, 62, 63, 64, 65, 66, 71, 83
377	1	10	8	59, 136, 137, 138, 145, 147, 150, 179
378	2	5, 10	9	93, 94, 95, 100, 102, 104, 106, 109, 114
379	1	10	10	86, 87, 88, 89, 90, 91, 92, 93, 94, 97
380	2	5, 10	10	46, 121, 122, 123, 124, 125, 126, 127, 131, 141
381	1	10	10	60, 165, 166, 167, 168, 169, 170, 172, 175, 179
382	2	2, 5	10	4, 47, 166, 171, 174, 177, 182, 187, 194, 199
383	1	1	7	63, 64, 65, 66, 68, 71, 83
384	1	1	10	23, 47, 142, 143, 144, 145, 147, 149, 151, 153
385	1	1	9	112, 113, 114, 115, 116, 117, 118, 119, 179
386	1	5	9	48, 154, 155, 156, 158, 159, 160, 162, 180
387	1	1	10	74, 75, 77, 79, 82, 85, 88, 91, 94, 97
388	2	2, 5	10	48, 136, 137, 138, 139, 140, 141, 156, 163, 186
389	1	1	10	62, 63, 64, 65, 66, 67, 68, 69, 71, 83
390	1	1	8	84, 85, 86, 87, 94, 97, 100, 103
391	1	5	9	84, 115, 116, 117, 118, 119, 120, 121, 122
392	1	1	9	17, 18, 19, 20, 21, 24, 27, 30, 42
393	1	1	9	53, 55, 57, 59, 61, 63, 68, 73, 78
394	2	2, 5	9	56, 57, 58, 59, 60, 61, 62, 63, 84
395	1	1	9	169, 170, 171, 172, 173, 174, 175, 176, 180
396	1	1	10	84, 169, 170, 171, 172, 173, 174, 175, 176, 187
397	3	1, 2, 5	7	51, 54, 57, 60, 69, 88, 114
398	1	1	9	21, 23, 24, 25, 27, 29, 31, 33, 85
399	2	2, 5	10	43, 44, 45, 46, 47, 48, 49, 50, 51, 85
400	1	5	9	85, 94, 95, 96, 97, 98, 99, 107, 109

Fonte: Elaborado pelo autor.

Tabela 27: Banco de dados da Instância No. 2.

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
1	3	3, 7, 8	10	3, 142, 144, 146, 148, 150, 152, 160, 164, 184
2	1	8	12	34, 35, 36, 39, 41, 43, 45, 47, 61, 64, 99, 171
3	1	10	12	91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 102, 172
4	2	2, 5	2	75, 76
5	8	1, 4, 5, 6, 7, 8, 9, 10	11	29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 101
6	2	4, 7	11	29, 30, 31, 32, 33, 34, 35, 36, 37, 40, 108
7	2	5, 6	10	2, 10, 14, 18, 22, 109, 194, 195, 198, 200
8	2	5, 10	7	28, 29, 30, 31, 32, 41, 110
9	1	1	11	144, 146, 148, 150, 152, 154, 156, 158, 162, 164, 172
10	3	3, 7, 8	10	95, 96, 97, 98, 99, 100, 102, 104, 105, 173
11	1	6	12	88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 102, 173
12	1	6	12	41, 42, 43, 44, 45, 46, 47, 48, 53, 63, 107, 174
13	2	5, 9	10	63, 64, 65, 66, 67, 68, 69, 71, 73, 110
14	4	4, 6, 8, 10	12	15, 17, 19, 21, 23, 25, 27, 29, 31, 41, 105, 177
15	1	5	11	110, 111, 112, 113, 116, 119, 122, 125, 128, 134, 148
16	2	4, 8	12	108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 178
17	1	10	12	125, 126, 127, 128, 129, 130, 131, 132, 133, 144, 148, 178
18	1	5	8	108, 112, 113, 114, 115, 116, 117, 118
19	1	2	2	81, 82
20	1	2	2	156, 157
21	4	3, 5, 7, 9	10	16, 18, 20, 22, 24, 26, 28, 30, 32, 106
22	2	1, 5	7	5, 6, 7, 100, 198, 199, 200
23	1	9	11	150, 151, 152, 153, 154, 155, 157, 159, 161, 163, 165
24	2	5, 6	6	108, 167, 168, 169, 170, 171
25	1	3	9	101, 102, 103, 105, 106, 107, 108, 109, 110
26	3	3, 7, 8	10	35, 36, 37, 38, 40, 42, 44, 48, 53, 100
27	9	1, 3, 4, 5, 6, 7, 8, 9, 10	10	18, 19, 20, 21, 22, 23, 32, 34, 42, 100
28	1	5	11	43, 44, 45, 46, 47, 48, 49, 50, 53, 63, 100
29	2	5, 10	7	56, 62, 65, 68, 72, 76, 100
30	2	4, 7	11	64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 101
31	1	4	9	132, 134, 136, 138, 140, 152, 156, 160, 164
32	1	1	9	43, 44, 45, 46, 47, 49, 53, 70, 106
33	5	1, 5, 7, 8, 9	11	101, 165, 166, 167, 168, 169, 170, 177, 179, 181, 183
34	2	5, 6	11	32, 33, 34, 35, 36, 37, 38, 39, 40, 71, 107
35	3	3, 7, 8	7	21, 23, 25, 27, 29, 33, 63
36	1	1	10	87, 88, 89, 90, 91, 92, 93, 94, 95, 96
37	4	1, 5, 9, 10	7	44, 46, 48, 49, 50, 64, 103
38	1	9	11	79, 80, 82, 84, 86, 88, 90, 92, 94, 105, 111
39	1	4	11	143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 154
40	1	5	11	41, 42, 43, 44, 46, 48, 50, 52, 54, 74, 102
41	6	1, 3, 5, 7, 8, 9	4	19, 20, 31, 101
42	1	2	2	11, 12
43	1	2	2	85, 86
44	2	3, 7	6	18, 23, 24, 28, 30, 101
45	1	1	11	1, 2, 3, 4, 5, 108, 196, 197, 198, 199, 200

Fonte: Elaborado pelo autor.

Tabela 27: Banco de dados da Instância No. 2 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
46	2	5, 10	11	128, 130, 132, 137, 140, 143, 146, 149, 152, 155, 158
47	2	2, 9	0	
48	1	1	7	26, 27, 28, 29, 30, 64, 102
49	2	5, 10	11	1, 3, 6, 10, 101, 195, 196, 197, 198, 199, 200
50	2	3, 7	6	164, 165, 166, 168, 172, 174
51	3	1, 3, 8	5	172, 173, 174, 175, 177
52	2	5, 6	11	112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122
53	1	9	11	79, 80, 81, 82, 83, 84, 85, 86, 87, 89, 101
54	1	7	11	108, 110, 113, 116, 119, 122, 126, 130, 134, 138, 148
55	1	9	10	1, 5, 9, 102, 174, 177, 179, 181, 188, 191
56	3	3, 7, 8	6	8, 10, 119, 197, 198, 199
57	1	1	10	12, 14, 16, 24, 27, 30, 33, 36, 65, 102
58	1	4	11	102, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137
59	1	3	8	15, 17, 18, 19, 20, 21, 22, 23
60	1	5	11	129, 131, 133, 138, 141, 144, 147, 150, 153, 156, 159
61	1	3	10	103, 104, 105, 106, 107, 108, 109, 110, 111, 112
62	9	1, 3, 4, 5, 6, 7, 8, 9, 10	9	14, 15, 16, 17, 18, 19, 20, 38, 102
63	2	3, 8	10	180, 181, 182, 183, 184, 185, 187, 189, 191, 194
64	8	1, 4, 5, 6, 7, 8, 9, 10	11	4, 102, 192, 193, 194, 195, 196, 197, 198, 199, 200
65	2	1, 5	4	145, 146, 148, 160
66	1	3	4	84, 85, 86, 87
67	1	9	10	158, 159, 160, 161, 162, 164, 165, 166, 167, 168
68	1	9	11	15, 113, 188, 189, 190, 191, 192, 193, 194, 195, 196
69	1	5	2	6, 103
70	2	3, 7	7	3, 4, 5, 7, 103, 197, 198
71	2	1, 8	11	134, 135, 136, 137, 138, 139, 140, 141, 142, 145, 149
72	1	4	10	2, 6, 10, 183, 184, 185, 186, 187, 189, 198
73	1	1	8	109, 110, 111, 114, 116, 117, 118, 119
74	3	1, 5, 6	11	127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 165
75	1	3	2	192, 195
76	2	1, 5	9	176, 177, 178, 179, 180, 181, 182, 183, 184
77	1	8	11	67, 69, 71, 74, 79, 82, 85, 88, 91, 94, 103
78	1	9	9	133, 137, 139, 141, 153, 157, 161, 165, 173
79	3	1, 3, 8	10	108, 109, 110, 111, 112, 113, 114, 115, 116, 117
80	1	9	8	128, 130, 132, 133, 134, 135, 147, 158
81	1	1	11	105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115
82	2	1, 8	10	95, 96, 97, 98, 99, 100, 102, 103, 104, 109
83	4	1, 5, 9, 10	4	137, 139, 141, 143
84	1	9	11	7, 8, 9, 10, 12, 19, 22, 32, 36, 66, 103
85	3	3, 7, 8	10	55, 56, 57, 59, 60, 61, 63, 65, 67, 112
86	1	3	6	140, 142, 143, 144, 145, 146
87	3	5, 7, 9	11	1, 2, 3, 7, 103, 195, 196, 197, 198, 199, 200
88	1	9	7	94, 95, 96, 97, 98, 100, 103,
89	1	9	11	1, 2, 3, 4, 8, 107, 196, 197, 198, 199, 200
90	1	5	11	11, 12, 13, 14, 15, 16, 17, 18, 19, 25, 109

Fonte: Elaborado pelo autor.

Tabela 27: Banco de dados da Instância No. 2 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
91	1	3	8	49, 51, 53, 54, 55, 56, 57, 58
92	1	2	2	1, 200
93	3	3, 7, 8	10	19, 20, 21, 22, 23, 24, 25, 26, 27, 28
94	1	4	4	4, 5, 6, 7
95	2	3, 7	10	167, 168, 169, 171, 172, 173, 174, 175, 176, 177
96	1	4	3	126, 127, 167
97	4	4, 5, 6, 10	11	2, 9, 108, 185, 186, 188, 190, 192, 194, 196, 198
98	1	1	10	41, 43, 45, 47, 49, 56, 59, 62, 65, 68
99	2	5, 6	11	149, 150, 151, 152, 153, 154, 155, 156, 157, 159, 162
100	2	5, 9	11	88, 89, 90, 91, 92, 93, 94, 95, 96, 98, 104
101	1	3	10	17, 18, 22, 24, 26, 28, 30, 32, 34, 67
102	4	4, 5, 6, 10	11	32, 33, 34, 35, 38, 39, 40, 41, 42, 67, 104
103	1	3	10	126, 127, 128, 129, 130, 131, 132, 133, 134, 135
104	2	3, 7	10	117, 118, 119, 120, 122, 123, 124, 143, 146, 150
105	3	1, 2, 5	2	166, 167
106	1	4	11	104, 171, 172, 173, 174, 175, 176, 178, 180, 182, 194
107	2	5, 9	11	8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 124
108	2	1, 8	11	49, 50, 51, 52, 53, 54, 62, 64, 66, 68, 114
109	2	3, 9	5	73, 74, 76, 77, 105
110	1	4	11	87, 88, 89, 90, 91, 92, 93, 94, 103, 105, 108
111	2	5, 6	11	104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114
112	4	4, 5, 6, 10	10	115, 116, 117, 119, 121, 123, 125, 127, 129, 134
113	1	4	11	38, 39, 42, 44, 46, 48, 51, 54, 57, 70, 105
114	1	8	11	106, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187
115	1	9	8	140, 141, 142, 143, 144, 148, 149, 150
116	1	8	11	73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 149
117	2	5, 6	3	125, 127, 151
118	2	5, 6	10	164, 165, 166, 167, 168, 169, 170, 171, 172, 174
119	2	2, 5	2	27, 28
120	1	1	11	158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168
121	1	2	2	192, 193
122	1	4	10	3, 11, 15, 19, 26, 111, 186, 187, 190, 192
123	2	3, 7	7	54, 55, 59, 60, 61, 62, 106
124	1	8	10	66, 67, 68, 74, 76, 78, 80, 82, 84, 110
125	4	1, 5, 9, 10	3	172, 173, 196
126	1	5	11	45, 46, 47, 48, 49, 50, 51, 52, 54, 68, 107
127	1	8	7	58, 59, 61, 70, 72, 74, 112
128	1	1	9	146, 147, 148, 149, 154, 158, 162, 166, 170
129	2	4, 5	5	48, 50, 54, 68, 107
130	2	1, 8	11	76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 117
131	2	3, 4	10	60, 61, 63, 65, 67, 75, 79, 83, 91, 111
132	2	2, 9	2	136, 149
133	3	3, 4, 8	8	145, 147, 149, 151, 153, 155, 161, 166
134	4	1, 5, 9, 10	11	131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 149
135	1	5	11	3, 10, 109, 184, 187, 189, 191, 193, 195, 197, 199

Fonte: Elaborado pelo autor.

Tabela 27: Banco de dados da Instância No. 2 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
136	1	3	10	2, 11, 190, 192, 194, 195, 196, 197, 198, 199
137	1	9	11	2, 4, 68, 150, 188, 190, 192, 194, 196, 198, 200
138	1	2	2	75, 76
139	1	4	10	43, 44, 45, 46, 47, 48, 49, 50, 69, 111
140	3	1, 5, 8	11	111, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195
141	3	3, 4, 7	7	24, 26, 27, 28, 29, 90, 112
142	1	9	11	125, 126, 127, 128, 129, 130, 131, 132, 133, 135, 150
143	1	9	11	107, 111, 114, 117, 120, 123, 127, 131, 135, 139, 150
144	4	1, 5, 9, 10	4	101, 102, 103, 104
145	3	1, 5, 6	11	164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174
146	1	2	2	169, 170
147	1	3	8	123, 125, 131, 133, 135, 137, 139, 141
148	3	3, 4, 7	3	14, 16, 120
149	2	5, 6	7	24, 25, 31, 33, 35, 77, 116
150	1	3	6	41, 42, 43, 44, 69, 77
151	1	3	7	13, 25, 28, 31, 34, 37, 69
152	4	4, 5, 6, 10	9	115, 179, 180, 181, 182, 184, 190, 192, 195
153	4	4, 5, 6, 10	11	5, 6, 7, 8, 9, 10, 11, 12, 13, 17, 115
154	1	3	10	135, 136, 139, 141, 143, 145, 153, 156, 159, 162
155	5	3, 4, 5, 7, 9	9	79, 81, 82, 83, 84, 85, 86, 87, 92
156	1	5	11	74, 75, 78, 79, 80, 81, 83, 85, 87, 92, 116
157	1	1	10	36, 37, 38, 39, 40, 41, 42, 43, 57, 69
158	2	5, 6	5	146, 147, 148, 149, 163
159	2	3, 7	10	4, 7, 11, 182, 184, 186, 188, 190, 192, 194
160	1	3	9	56, 57, 58, 59, 60, 61, 62, 63, 78
161	1	1	11	24, 25, 26, 27, 28, 29, 30, 33, 38, 70, 123
162	3	3, 4, 7	8	126, 127, 132, 136, 138, 140, 142, 151
163	1	5	8	117, 118, 119, 120, 121, 122, 123, 124
164	1	5	11	156, 157, 158, 159, 160, 161, 162, 163, 164, 167, 168
165	1	2	2	93, 94
166	2	1, 8	11	123, 141, 142, 146, 149, 152, 155, 158, 170, 174, 178
167	4	4, 5, 7, 9	11	18, 116, 167, 168, 169, 170, 171, 172, 173, 174, 175
168	1	9	11	134, 135, 136, 139, 142, 145, 148, 151, 154, 157, 160
169	1	9	5	60, 63, 66, 70, 119
170	4	1, 5, 9, 10	11	102, 103, 104, 105, 106, 124, 128, 132, 136, 141, 151
171	1	8	11	104, 106, 108, 114, 117, 120, 123, 126, 129, 135, 151
172	1	9	11	146, 147, 148, 149, 150, 151, 154, 157, 160, 163, 165
173	2	1, 8	11	175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185
174	2	2, 5	2	58, 59
175	1	4	11	97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 117
176	1	5	3	134, 135, 136
177	3	3, 7, 8	10	173, 174, 175, 176, 177, 178, 179, 180, 181, 182
178	1	9	4	153, 154, 165, 168
179	1	5	11	45, 46, 47, 48, 49, 50, 57, 59, 61, 71, 124
180	1	4	8	34, 35, 40, 49, 52, 55, 58, 70

Fonte: Elaborado pelo autor.

Tabela 27: Banco de dados da Instância No. 2 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
181	1	3	10	113, 114, 115, 116, 117, 118, 119, 120, 121, 122
182	1	5	9	58, 59, 60, 61, 62, 64, 65, 66, 70
183	3	1, 5, 6	11	119, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180
184	2	1, 5	11	137, 138, 139, 140, 141, 142, 143, 145, 147, 149, 151
185	1	5	10	39, 40, 41, 42, 45, 53, 55, 57, 71, 119
186	1	3	10	40, 41, 42, 43, 44, 45, 47, 49, 65, 71
187	3	1, 5, 6	9	142, 143, 144, 145, 151, 155, 159, 163, 175
188	2	4, 7	11	31, 32, 34, 35, 36, 37, 38, 39, 40, 85, 120
189	3	1, 2, 5	2	165, 166
190	2	5, 10	10	128, 129, 130, 131, 132, 133, 134, 135, 136, 137
191	1	2	2	47, 48
192	2	5, 6	0	
193	1	1	11	89, 90, 91, 93, 95, 97, 99, 101, 103, 105, 120
194	2	5, 10	11	14, 15, 16, 17, 18, 19, 20, 21, 22, 26, 121
195	4	3, 5, 7, 9	10	84, 85, 86, 87, 88, 89, 90, 91, 92, 93
196	1	8	3	13, 14, 164
197	1	1	10	48, 49, 50, 51, 52, 54, 56, 58, 72, 120
198	2	3, 8	10	84, 85, 86, 87, 88, 89, 90, 91, 92, 93
199	2	7, 8	11	40, 41, 43, 45, 47, 49, 52, 55, 67, 71, 120
200	4	4, 5, 6, 10	11	68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 121
201	4	1, 5, 6, 8	11	68, 70, 72, 77, 80, 83, 86, 89, 92, 96, 121
202	1	5	6	1, 2, 71, 152, 199, 200
203	1	3	10	179, 180, 181, 182, 183, 184, 185, 186, 187, 188
204	1	7	11	124, 160, 161, 162, 163, 164, 165, 168, 171, 175, 191
205	1	4	11	33, 34, 36, 38, 40, 42, 44, 50, 59, 72, 121
206	1	4	11	38, 39, 40, 45, 47, 49, 51, 53, 59, 75, 121
207	1	2	2	58, 59
208	1	9	11	12, 121, 189, 190, 191, 192, 193, 194, 195, 196, 197
209	1	1	11	65, 66, 73, 78, 81, 84, 87, 90, 93, 97, 122
210	3	1, 5, 9	10	29, 122, 174, 175, 176, 177, 178, 179, 181, 185
211	1	3	10	105, 107, 109, 115, 118, 121, 124, 127, 130, 136
212	3	5, 7, 9	11	145, 146, 147, 148, 149, 156, 158, 160, 162, 164, 166
213	2	5, 10	6	20, 26, 27, 28, 29, 121
214	1	7	11	22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 125
215	1	4	11	85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 152
216	2	5, 9	11	48, 49, 50, 51, 52, 53, 54, 55, 60, 72, 122
217	2	5, 6	11	126, 127, 128, 129, 130, 133, 136, 139, 142, 145, 152
218	4	3, 4, 7, 8	8	36, 37, 41, 50, 53, 56, 60, 72
219	2	2, 5	2	30, 33
220	4	3, 5, 7, 9	8	51, 52, 53, 54, 55, 56, 57, 58
221	2	7, 8	11	21, 22, 23, 24, 25, 26, 27, 28, 29, 72, 122
222	1	2	2	37, 38
223	1	7	11	107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 123
224	2	5, 10	11	93, 122, 175, 176, 177, 178, 179, 186, 188, 190, 192
225	2	5, 10	10	156, 157, 158, 159, 161, 162, 163, 164, 166, 169

Fonte: Elaborado pelo autor.

Tabela 27: Banco de dados da Instância No. 2 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
226	2	5, 6	11	137, 138, 140, 142, 144, 152, 155, 158, 161, 164, 166
227	3	5, 7, 9	11	27, 122, 176, 177, 178, 179, 180, 181, 182, 183, 186
228	2	2, 9	2	76, 77
229	1	5	11	52, 53, 54, 55, 56, 57, 58, 59, 60, 78, 122
230	4	1, 5, 6, 8	8	27, 28, 29, 30, 31, 32, 39, 123
231	2	5, 10	11	81, 82, 83, 84, 86, 88, 90, 92, 94, 96, 123
232	1	9	11	28, 125, 144, 145, 147, 150, 153, 156, 159, 172, 176
233	2	5, 10	11	12, 13, 14, 15, 16, 17, 25, 27, 29, 73, 123
234	2	2, 5	0	
235	2	1, 5	10	55, 56, 57, 60, 61, 62, 64, 66, 73, 123
236	1	3	10	14, 15, 16, 17, 18, 20, 23, 34, 39, 73
237	1	5	10	1, 4, 28, 123, 185, 187, 189, 191, 193, 195
238	2	4, 5	11	29, 124, 185, 186, 187, 188, 189, 190, 191, 192, 193
239	2	5, 10	11	95, 96, 97, 98, 99, 100, 101, 102, 104, 106, 124
240	1	8	11	33, 34, 35, 36, 37, 50, 53, 56, 68, 73, 124
241	1	3	5	45, 46, 47, 50, 61
242	3	4, 7, 8	11	75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 125
243	1	1	11	113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 126
244	4	4, 5, 6, 10	10	16, 17, 18, 19, 20, 21, 22, 23, 24, 157
245	1	5	10	59, 60, 62, 63, 64, 65, 66, 69, 74, 125
246	2	3, 4	7	141, 142, 143, 144, 145, 160, 165
247	1	9	3	138, 140, 142
248	2	3, 7	10	55, 56, 57, 58, 68, 80, 84, 88, 94, 124
249	2	3, 4	8	6, 7, 9, 10, 11, 13, 14, 83
250	1	1	10	96, 97, 98, 99, 100, 101, 102, 103, 104, 124
251	4	3, 5, 7, 9	8	30, 31, 34, 36, 38, 44, 64, 74
252	4	1, 5, 6, 8	11	123, 124, 127, 129, 131, 133, 135, 137, 139, 146, 152
253	2	3, 7	7	45, 47, 49, 51, 61, 78, 125
254	1	5	11	121, 122, 123, 124, 131, 134, 137, 140, 143, 146, 153
255	1	3	10	60, 62, 64, 66, 68, 70, 72, 74, 76, 78
256	1	8	6	75, 76, 77, 78, 80, 125
257	4	3, 4, 7, 8	10	1, 2, 3, 4, 75, 169, 197, 198, 199, 200
258	1	3	10	105, 106, 109, 111, 113, 115, 117, 119, 121, 123
259	4	4, 5, 6, 10	11	4, 8, 12, 16, 20, 75, 125, 188, 189, 191, 193
260	1	3	10	150, 151, 152, 153, 154, 155, 156, 157, 158, 159
261	2	2, 5	2	55, 56
262	1	4	11	152, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183
263	1	9	11	42, 44, 46, 48, 54, 57, 60, 63, 66, 94, 126
264	2	1, 5	11	139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 153
265	3	2, 3, 9	2	176, 178
266	1	4	11	51, 52, 53, 54, 55, 56, 58, 60, 62, 85, 126
267	1	8	10	79, 126, 184, 185, 186, 187, 188, 189, 190, 191
268	2	4, 8	11	63, 64, 65, 66, 67, 68, 69, 70, 71, 75, 126
269	8	1, 4, 5, 6, 7, 8, 9, 10	11	110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 126
270	1	8	11	75, 127, 162, 163, 164, 165, 166, 167, 174, 184, 187

Fonte: Elaborado pelo autor.

Tabela 27: Banco de dados da Instância No. 2 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
271	1	5	11	75, 127, 140, 143, 148, 151, 154, 157, 169, 173, 177
272	1	2	2	171, 172
273	2	5, 6	10	131, 132, 133, 134, 136, 144, 146, 148, 150, 153
274	1	2	2	182, 183
275	1	1	11	125, 126, 128, 130, 132, 134, 136, 138, 140, 147, 153
276	1	2	2	76, 77
277	2	5, 10	9	33, 34, 35, 36, 37, 38, 39, 76, 154
278	4	1, 5, 9, 10	9	69, 70, 71, 73, 74, 75, 76, 79, 127
279	1	3	10	76, 168, 169, 170, 171, 172, 173, 175, 185, 188
280	4	1, 2, 5, 8	2	73, 74
281	1	5	8	167, 183, 184, 185, 186, 187, 188, 189
282	1	3	10	113, 114, 118, 120, 122, 124, 126, 128, 130, 138
283	2	5, 9	10	59, 62, 64, 66, 73, 77, 81, 85, 89, 153
284	3	1, 5, 8	5	57, 60, 63, 74, 156
285	1	2	2	153, 154
286	1	8	11	65, 66, 67, 68, 69, 70, 71, 72, 73, 79, 154
287	1	2	2	50, 52
288	3	1, 5, 8	8	3, 7, 76, 178, 180, 182, 190, 199
289	2	5, 6	11	13, 14, 15, 16, 17, 18, 19, 20, 21, 77, 154
290	4	1, 5, 9, 10	11	112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 167
291	4	1, 5, 9, 10	9	76, 136, 137, 138, 139, 140, 141, 161, 186
292	1	9	4	137, 138, 139, 141
293	1	4	11	53, 54, 55, 56, 57, 58, 59, 60, 62, 81, 168
294	1	9	10	61, 62, 63, 64, 65, 66, 67, 69, 77, 155
295	2	5, 10	11	109, 112, 115, 118, 121, 125, 129, 133, 137, 147, 155
296	1	1	11	7, 8, 9, 10, 11, 12, 13, 14, 23, 77, 155
297	1	8	11	157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167
298	2	2, 5	2	36, 37
299	1	8	7	42, 58, 61, 64, 67, 77, 161
300	1	9	11	79, 157, 188, 189, 190, 191, 192, 193, 194, 195, 196
301	1	9	11	3, 4, 5, 6, 7, 8, 9, 10, 11, 80, 168
302	1	8	3	84, 95, 155
303	1	8	4	172, 174, 176, 178
304	2	1, 5	11	108, 109, 110, 111, 112, 114, 116, 118, 120, 122, 155
305	2	5, 6	11	23, 24, 25, 26, 27, 28, 29, 30, 31, 78, 155
306	2	1, 5	11	82, 83, 84, 85, 86, 87, 88, 89, 90, 94, 163
307	1	2	2	129, 131
308	1	8	11	118, 119, 120, 125, 132, 135, 138, 141, 144, 147, 156
309	1	2	2	85, 86
310	3	3, 4, 7	7	32, 33, 35, 37, 43, 65, 78
311	1	1	11	31, 32, 33, 40, 42, 44, 46, 51, 63, 82, 169
312	1	5	11	23, 24, 25, 26, 27, 28, 29, 30, 31, 78, 156
313	1	8	11	4, 5, 6, 11, 13, 21, 31, 35, 40, 86, 156
314	2	3, 4	10	8, 79, 174, 175, 176, 177, 178, 179, 180, 181
315	2	1, 8	11	37, 38, 39, 40, 41, 47, 49, 51, 61, 80, 161

Fonte: Elaborado pelo autor.

Tabela 27: Banco de dados da Instância No. 2 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
316	1	5	11	78, 81, 83, 85, 87, 89, 91, 93, 95, 106, 156
317	1	4	11	27, 28, 29, 30, 31, 32, 33, 34, 35, 80, 157
318	1	2	2	191, 193
319	1	9	11	87, 88, 92, 94, 96, 98, 100, 102, 104, 106, 158
320	2	1, 8	11	50, 51, 52, 53, 55, 58, 61, 64, 67, 95, 157
321	2	1, 5	6	88, 89, 90, 91, 94, 179
322	2	5, 9	8	157, 179, 180, 181, 182, 183, 186, 187
323	1	8	10	61, 63, 65, 67, 69, 71, 73, 75, 77, 80
324	1	2	2	194, 195
325	4	1, 3, 4, 8	8	126, 127, 131, 132, 133, 134, 138, 170
326	1	3	10	137, 138, 139, 140, 141, 142, 143, 144, 145, 147
327	2	1, 5	11	5, 10, 15, 83, 158, 179, 180, 181, 184, 187, 190
328	1	2	2	43, 44
329	2	5, 9	11	56, 57, 58, 59, 60, 61, 62, 63, 66, 81, 158
330	3	1, 5, 8	11	107, 108, 110, 112, 114, 116, 118, 120, 122, 124, 162
331	1	4	10	2, 5, 98, 159, 186, 188, 190, 192, 194, 196
332	1	3	10	6, 7, 8, 9, 10, 11, 12, 13, 14, 107
333	1	2	2	189, 190
334	1	8	9	32, 33, 34, 39, 41, 43, 45, 52, 81
335	1	8	9	86, 87, 88, 89, 90, 91, 92, 93, 94
336	4	4, 5, 6, 10	10	117, 118, 119, 120, 121, 122, 123, 124, 1
337	2	4, 8	11	1, 2, 3, 4, 5, 106, 159, 197, 198, 199, 200
338	1	4	11	147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 172
339	1	5	6	59, 62, 65, 68, 81, 162
340	1	5	10	2, 5, 9, 81, 183, 185, 187, 189, 191, 193
341	4	4, 5, 6, 10	10	5, 6, 7, 8, 9, 10, 11, 12, 13, 81
342	2	7, 8	11	32, 35, 37, 39, 41, 43, 46, 52, 62, 82, 169
343	1	2	2	14, 15
344	1	3	4	6, 7, 15, 82
345	1	9	11	83, 170, 189, 190, 191, 192, 193, 194, 195, 196, 197
346	1	1	10	19, 20, 21, 22, 23, 24, 25, 26, 27, 83
347	2	5, 6	11	32, 33, 34, 35, 36, 37, 38, 39, 40, 84, 160
348	2	5, 6	10	21, 22, 23, 24, 25, 26, 37, 40, 95, 168
349	2	5, 6	9	82, 160, 176, 177, 178, 183, 189, 191, 193
350	8	1, 4, 5, 6, 7, 8, 9, 10	11	78, 79, 80, 85, 87, 89, 91, 93, 95, 97, 160
351	1	5	11	136, 137, 138, 139, 140, 141, 142, 143, 144, 148, 161
352	1	2	1	95
353	2	5, 6	11	69, 70, 71, 72, 73, 82, 84, 86, 88, 96, 161
354	8	1, 4, 5, 6, 7, 8, 9, 10	11	104, 105, 106, 107, 113, 115, 117, 119, 121, 123, 169
355	1	9	9	17, 19, 20, 21, 22, 23, 24, 99, 161
356	3	1, 5, 6	9	60, 61, 62, 63, 64, 65, 66, 78, 82
357	4	4, 5, 6, 10	11	1, 3, 5, 83, 162, 189, 191, 193, 195, 197, 199
358	4	4, 5, 6, 10	7	38, 39, 51, 54, 57, 62, 84
359	1	5	10	72, 73, 74, 75, 76, 77, 78, 79, 83, 170
360	2	5, 6	7	55, 58, 61, 64, 67, 96, 162

Fonte: Elaborado pelo autor.

Tabela 27: Banco de dados da Instância No. 2 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
361	2	7, 8	11	125, 126, 127, 128, 129, 130, 131, 132, 133, 153, 170
362	2	1, 5	10	1, 5, 9, 13, 17, 21, 163, 196, 197, 199
363	1	1	9	30, 31, 32, 33, 34, 35, 36, 96, 170
364	1	4	11	51, 52, 53, 54, 55, 56, 57, 58, 62, 96, 163
365	2	2, 5	1	6
366	3	2, 3, 9	2	1, 200
367	2	2, 5	2	142, 143
368	1	3	10	6, 11, 16, 98, 175, 176, 182, 185, 188, 191
369	1	5	11	146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 163
370	1	3	7	99, 101, 102, 104, 105, 106, 107
371	1	3	9	70, 72, 73, 75, 77, 79, 81, 83, 96
372	1	3	10	77, 79, 81, 83, 85, 87, 89, 91, 93, 97
373	1	3	3	3, 6, 197
374	4	1, 5, 6, 8	11	1, 2, 3, 4, 5, 6, 7, 9, 107, 163, 200
375	1	4	11	96, 97, 98, 99, 100, 101, 102, 103, 104, 108, 163
376	1	4	9	97, 154, 155, 156, 157, 158, 159, 162, 187
377	3	1, 4, 8	11	42, 43, 44, 45, 46, 48, 50, 52, 63, 97, 175
378	1	3	5	79, 80, 83, 97, 99
379	1	4	11	1, 3, 7, 12, 17, 99, 164, 197, 198, 199, 200
380	2	3, 4	10	48, 49, 50, 51, 52, 53, 54, 55, 74, 97
381	3	3, 4, 8	8	23, 26, 29, 32, 35, 38, 97, 165
382	1	5	10	10, 11, 12, 14, 16, 18, 20, 22, 24, 98
383	1	5	11	2, 4, 8, 13, 18, 100, 166, 193, 194, 195, 196
384	1	5	11	1, 2, 3, 4, 5, 6, 7, 8, 9, 98, 166
385	1	5	6	9, 12, 13, 14, 15, 16
386	1	5	9	64, 67, 68, 69, 70, 71, 72, 100, 173
387	1	5	9	184, 185, 186, 187, 188, 189, 190, 191, 193
388	1	5	4	98, 180, 182, 188
389	1	5	9	69, 70, 71, 72, 74, 78, 82, 90, 107
390	2	5, 6	11	15, 16, 17, 18, 19, 20, 21, 22, 25, 101, 171
391	2	5, 6	9	8, 9, 13, 15, 17, 19, 21, 23, 25
392	2	5, 6	9	35, 36, 37, 38, 39, 46, 52, 98, 169
393	2	5, 6	10	99, 179, 180, 181, 182, 183, 184, 185, 186, 187
394	2	5, 6	11	85, 86, 87, 88, 89, 90, 91, 92, 93, 99, 171
395	3	1, 5, 6	4	168, 173, 175, 177
396	9	1, 3, 4, 5, 6, 7, 8, 9, 10	10	1, 2, 109, 194, 195, 196, 197, 198, 199, 200
397	2	5, 6	10	99, 143, 145, 147, 149, 151, 153, 163, 183, 198
398	8	1, 4, 5, 6, 7, 8, 9, 10	11	9, 14, 30, 101, 170, 177, 178, 183, 186, 189, 192
399	4	3, 5, 7, 9	7	95, 96, 97, 98, 99, 100, 176
400	1	7	11	4, 8, 99, 171, 192, 193, 194, 195, 196, 197, 200

Fonte: Elaborado pelo autor.

Tabela 28: Banco de dados da Instância No. 3.

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
1	1	8	9	23, 24, 25, 26, 28, 30, 32, 34, 134
2	8	1, 3, 4, 5, 6, 7, 8, 10	11	122, 124, 126, 128, 130, 132, 134, 136, 138, 169, 198
3	2	3, 7	11	119, 120, 121, 122, 123, 124, 125, 127, 129, 131, 198
4	1	5	7	157, 158, 159, 160, 161, 164, 165
5	1	2	5	62, 78, 82, 127, 199
6	1	8	11	28, 29, 30, 31, 32, 38, 40, 42, 44, 91, 199
7	8	1, 3, 4, 5, 6, 7, 8, 10	11	168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 199
8	4	4, 5, 7, 8	11	15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 200
9	1	8	9	43, 45, 47, 49, 51, 53, 55, 89, 200
10	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	79, 80, 81, 82, 83, 84, 85, 86, 87, 90
11	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	162, 163, 164, 165, 166, 167, 168, 169, 170, 173
12	1	5	10	150, 151, 152, 153, 154, 155, 156, 157, 158, 159
13	3	1, 4, 5,	11	155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 200
14	1	7	11	1, 53, 54, 55, 56, 58, 61, 76, 81, 86, 92
15	1	5	11	1, 135, 136, 138, 141, 144, 147, 150, 153, 156, 159
16	1	8	10	1, 4, 7, 16, 187, 188, 189, 192, 195, 198
17	1	5	11	1, 2, 4, 193, 194, 195, 196, 197, 198, 199, 200
18	1	9	10	120, 121, 122, 124, 128, 132, 136, 140, 145, 150
19	2	2, 9	9	1, 99, 100, 101, 102, 104, 112, 118, 127
20	1	4	10	74, 75, 76, 77, 78, 79, 80, 81, 82, 90
21	1	4	11	2, 88, 89, 90, 91, 92, 93, 94, 95, 96, 98
22	1	5	10	2, 28, 29, 30, 31, 32, 33, 34, 35, 36
23	2	3, 7	11	2, 164, 165, 166, 167, 168, 169, 176, 178, 180, 182
24	8	1, 3, 4, 5, 6, 7, 8, 10	11	3, 38, 39, 40, 42, 44, 46, 48, 50, 52, 90
25	3	1, 4, 5	10	150, 151, 152, 153, 154, 156, 159, 162, 165, 170
26	3	3, 7, 9	10	10, 11, 12, 13, 14, 15, 28, 31, 34, 90
27	1	2	6	4, 22, 23, 24, 25, 26
28	1	7	11	4, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20
29	2	4, 5	9	5, 96, 97, 100, 102, 108, 112, 120, 128
30	8	1, 3, 4, 5, 6, 7, 8, 10	11	5, 48, 49, 50, 51, 52, 53, 54, 55, 72, 91
31	1	9	10	66, 67, 68, 69, 70, 71, 72, 73, 82, 91
32	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	146, 147, 148, 149, 150, 151, 152, 153, 154, 156
33	1	2	10	145, 147, 149, 151, 153, 155, 157, 166, 169, 172
34	1	7	10	27, 28, 29, 30, 35, 37, 39, 41, 43, 91
35	1	9	5	32, 33, 34, 39, 91
36	3	3, 7, 9	10	6, 8, 10, 12, 14, 16, 18, 20, 22, 24
37	1	2	5	41, 49, 51, 53, 73
38	1	5	10	158, 159, 160, 161, 162, 163, 164, 167, 170, 173
39	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	119, 121, 124, 127, 135, 139, 143, 147, 151
40	1	8	10	1, 3, 16, 187, 188, 191, 193, 195, 197, 199

Fonte: Elaborado pelo autor.

Tabela 28: Banco de dados da Instância No. 3 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
41	3	3, 7, 9	10	133, 134, 135, 136, 137, 138, 139, 140, 141, 142
42	1	7	8	20, 21, 22, 23, 24, 25, 40, 93
43	1	4	10	7, 10, 13, 21, 195, 196, 197, 198, 199, 200
44	1	7	3	49, 53, 93
45	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	114, 116, 118, 125, 129, 133, 137, 141, 146, 160
46	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	132, 134, 139, 142, 145, 148, 151, 154, 157, 160
47	1	5	10	161, 162, 163, 164, 165, 166, 167, 168, 169, 170
48	1	8	2	55, 56
49	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	61, 62, 63, 64, 65, 66, 67, 68, 73, 99
50	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	5	2, 6, 21, 183, 189
51	1	7	9	3, 7, 22, 174, 175, 184, 187, 190, 199
52	3	1, 4, 5	10	18, 19, 20, 21, 22, 29, 31, 33, 40, 135
53	1	9	9	92, 93, 95, 97, 99, 101, 104, 118, 137
54	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	8	126, 127, 128, 129, 130, 131, 150, 162
55	1	7	10	122, 123, 124, 125, 126, 127, 128, 129, 130, 131
56	1	5	10	1, 3, 10, 13, 16, 19, 22, 195, 196, 197
57	1	5	3	154, 156, 157
58	1	4	10	53, 54, 55, 56, 58, 60, 63, 66, 119, 128
59	1	4	9	2, 5, 8, 23, 183, 190, 193, 196, 199
60	1	7	10	45, 46, 47, 48, 49, 61, 64, 81, 120, 129
61	1	7	10	2, 4, 11, 14, 17, 20, 23, 198, 199, 200
62	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	33, 34, 35, 36, 37, 38, 39, 47, 73, 94
63	3	3, 7, 9	10	168, 169, 170, 171, 172, 173, 174, 175, 176, 177
64	1	5	7	44, 46, 48, 50, 52, 54, 94
65	2	4, 5	10	43, 44, 45, 46, 49, 52, 55, 58, 77, 94
66	1	5	8	90, 92, 96, 98, 105, 110, 118, 136
67	1	5	8	3, 4, 5, 6, 7, 8, 9, 130
68	1	2	10	62, 63, 64, 65, 66, 72, 74, 77, 80, 95
69	3	1, 4, 5	10	77, 78, 79, 80, 81, 82, 83, 84, 85, 131
70	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	3	8, 23, 196
71	1	9	9	4, 8, 23, 193, 195, 196, 197, 198, 200
72	3	3, 7, 9	10	171, 172, 173, 174, 175, 176, 177, 178, 179, 180
73	1	7	10	116, 118, 120, 122, 124, 126, 128, 130, 140, 144
74	1	7	10	91, 92, 93, 97, 99, 102, 105, 114, 118, 163
75	1	8	10	106, 107, 108, 109, 110, 111, 112, 113, 118, 131
76	3	3, 7, 9	10	147, 148, 149, 150, 151, 152, 153, 154, 155, 166
77	1	4	8	129, 130, 131, 139, 141, 143, 145, 147
78	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	113, 114, 115, 116, 117, 118, 120, 129, 132, 135
79	1	9	10	98, 99, 100, 101, 102, 103, 105, 111, 119, 132
80	4	4, 5, 7, 8	10	49, 50, 51, 52, 53, 54, 55, 58, 75, 97
81	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	8	3, 6, 9, 24, 186, 188, 190, 192
82	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	46, 48, 50, 52, 59, 72, 77, 82, 87, 95
83	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	18, 19, 20, 21, 22, 23, 24, 25, 26
84	3	1, 4, 5	10	84, 85, 88, 90, 92, 94, 96, 98, 121, 132
85	1	4	9	125, 126, 127, 128, 129, 130, 131, 132, 133

Fonte: Elaborado pelo autor.

Tabela 28: Banco de dados da Instância No. 3 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
86	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	130, 131, 132, 133, 134, 136, 140, 144, 148, 152
87	5	1, 4, 5, 7, 8	10	112, 113, 119, 126, 130, 134, 138, 142, 147, 161
88	3	3, 7, 9	10	109, 110, 111, 112, 113, 114, 115, 116, 119, 132
89	1	7	10	17, 19, 21, 23, 25, 27, 29, 31, 33, 97
90	1	8	10	14, 15, 16, 17, 18, 19, 20, 21, 22, 24
91	3	3, 7, 9	9	99, 100, 101, 102, 104, 106, 111, 119, 141
92	3	1, 4, 5	10	82, 83, 84, 85, 86, 87, 88, 90, 92, 96
93	1	5	9	116, 122, 125, 128, 137, 141, 145, 149, 157
94	1	4	10	17, 18, 19, 20, 21, 37, 41, 45, 50, 133
95	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	59, 60, 61, 62, 63, 64, 65, 67, 74, 96
96	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	100, 101, 102, 103, 104, 105, 106, 112, 122, 133
97	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	1, 2, 3, 4, 5, 196, 197, 198, 199, 200
98	1	7	9	108, 109, 110, 111, 112, 113, 114, 119, 134
99	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	4	113, 115, 117, 118
100	1	9	10	74, 75, 76, 77, 78, 79, 80, 81, 82, 97
101	1	4	10	123, 124, 125, 126, 127, 128, 129, 130, 131, 143
102	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	148, 149, 150, 151, 152, 153, 154, 155, 156, 158
103	1	8	7	59, 63, 64, 71, 73, 75, 148
104	1	8	5	153, 154, 163, 168, 171
105	1	8	5	64, 79, 83, 119, 138
106	3	1, 4, 5	4	48, 59, 76, 134
107	1	2	10	123, 125, 127, 129, 131, 133, 135, 137, 139, 171
108	3	3, 7, 9	7	60, 61, 63, 80, 84, 120, 139
109	1	8	10	106, 107, 108, 109, 110, 111, 114, 117, 120, 144
110	1	9	3	35, 36, 159
111	3	3, 7, 9	4	54, 55, 75, 164
112	1	5	10	37, 39, 41, 47, 50, 53, 56, 59, 78, 135
113	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	6	164, 165, 166, 168, 169, 181
114	3	1, 4, 5	9	121, 122, 129, 132, 135, 139, 143, 148, 160
115	1	8	10	2, 3, 4, 5, 6, 7, 8, 9, 41, 135
116	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	105, 106, 107, 108, 109, 110, 111, 113, 120, 135
117	3	1, 4, 5	8	65, 67, 69, 70, 72, 74, 76, 149
118	1	9	10	12, 13, 14, 15, 16, 17, 18, 19, 20, 136
119	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	5	137, 138, 139, 140, 141
120	1	7	10	25, 26, 27, 28, 34, 38, 42, 46, 51, 136
121	1	7	10	171, 172, 173, 174, 175, 176, 177, 178, 179, 183
122	3	1, 4, 5	9	67, 68, 69, 70, 72, 84, 88, 93, 136
123	1	8	10	170, 171, 172, 173, 174, 175, 176, 177, 178, 179
124	1	2	10	76, 77, 78, 79, 80, 81, 89, 91, 93, 136
125	1	9	10	102, 103, 104, 105, 106, 107, 110, 113, 149, 161
126	1	4	6	95, 96, 97, 98, 109, 136
127	3	1, 4, 5	10	137, 138, 139, 140, 141, 142, 143, 144, 145, 161
128	1	5	10	22, 23, 24, 25, 26, 27, 28, 36, 41, 137
129	1	9	9	57, 59, 61, 63, 65, 67, 70, 75, 137
130	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	56, 58, 60, 62, 64, 66, 68, 71, 76, 138

Fonte: Elaborado pelo autor.

Tabela 28: Banco de dados da Instância No. 3 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
131	3	1, 4, 5	10	160, 161, 162, 164, 166, 168, 170, 172, 174, 176
132	3	3, 7, 9	10	50, 51, 52, 53, 54, 55, 56, 57, 75, 137
133	1	5	10	75, 76, 77, 78, 79, 80, 82, 84, 86, 162
134	1	7	6	63, 66, 73, 89, 94, 137
135	1	9	10	153, 154, 155, 156, 157, 158, 159, 160, 161, 162
136	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	77, 78, 79, 80, 81, 82, 83, 84, 85, 139
137	1	9	9	141, 142, 143, 144, 146, 147, 148, 149, 151
138	3	1, 4, 5	10	99, 100, 101, 102, 103, 104, 105, 113, 120, 147
139	1	2	10	174, 175, 176, 177, 178, 179, 180, 181, 182, 183
140	1	2	10	52, 53, 54, 55, 56, 62, 64, 67, 76, 162
141	1	9	10	8, 9, 10, 11, 12, 13, 14, 15, 16, 25
142	1	4	8	140, 141, 142, 143, 144, 150, 152, 161
143	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	138, 139, 140, 141, 142, 143, 144, 145, 146, 147
144	1	9	10	142, 143, 144, 145, 146, 147, 148, 149, 150, 162
145	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	8	90, 91, 92, 94, 98, 114, 120, 140
146	1	2	10	170, 171, 172, 173, 174, 175, 176, 177, 178, 179
147	1	4	10	175, 176, 177, 178, 183, 185, 187, 189, 191, 193
148	4	4, 5, 7, 8	4	12, 13, 17, 28
149	1	4	1	142
150	1	2	10	100, 102, 104, 106, 108, 110, 118, 121, 124, 163
151	1	4	10	5, 6, 7, 8, 9, 12, 15, 18, 21, 25
152	3	3, 7, 9	10	182, 183, 184, 185, 186, 187, 188, 189, 190, 191
153	1	2	8	16, 18, 20, 22, 26, 29, 32, 162
154	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	48, 49, 50, 51, 52, 53, 54, 55, 81
155	1	9	6	90, 91, 93, 94, 98, 167
156	1	4	10	109, 110, 111, 112, 113, 114, 115, 116, 121, 143
157	1	8	10	42, 43, 44, 45, 46, 47, 48, 49, 50, 144
158	1	4	10	33, 34, 35, 36, 37, 39, 41, 43, 45, 163
159	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	11, 16, 26, 176, 177, 182, 184, 186, 188
160	1	7	10	151, 152, 153, 154, 155, 156, 157, 158, 159, 163
161	2	2, 9	10	71, 72, 73, 74, 75, 77, 79, 82, 123, 144
162	1	9	10	147, 148, 149, 150, 151, 152, 166, 169, 172, 175
163	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	8	127, 130, 133, 136, 140, 144, 150, 164
164	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	62, 63, 64, 65, 66, 67, 69, 71, 94, 145
165	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	6	187, 188, 189, 190, 191, 192
166	3	1, 4, 5	4	133, 136, 140, 146
167	1	8	10	7, 9, 11, 13, 15, 17, 19, 21, 23, 27
168	1	9	10	10, 12, 16, 19, 22, 25, 34, 38, 42, 165
169	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	56, 57, 58, 59, 60, 61, 62, 63, 82, 164
170	3	1, 4, 5	10	1, 2, 3, 4, 5, 6, 8, 11, 14, 27
171	3	1, 4, 5	10	133, 137, 140, 143, 146, 149, 152, 155, 158, 165
172	3	1, 4, 5	10	11, 12, 13, 14, 15, 16, 17, 18, 19, 27
173	1	4	10	40, 41, 42, 43, 44, 45, 46, 56, 78, 165
174	1	5	10	3, 4, 5, 6, 7, 8, 9, 10, 11, 165
175	1	4	10	1, 2, 3, 4, 5, 6, 7, 166, 199, 200

Fonte: Elaborado pelo autor.

Tabela 28: Banco de dados da Instância No. 3 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
176	1	8	10	84, 85, 86, 87, 88, 89, 90, 115, 121, 166
177	1	5	9	147, 148, 149, 150, 151, 152, 153, 154, 184
178	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	74, 75, 76, 77, 78, 79, 80, 81, 82, 166
179	1	4,	4	150, 151, 152, 166
180	3	1, 4, 5	9	45, 46, 47, 48, 49, 50, 51, 52, 53
181	1	9	9	60, 61, 62, 63, 64, 65, 66, 68, 167
182	3	1, 4, 5	4	55, 56, 82, 171
183	1	4	10	104, 105, 106, 107, 108, 109, 110, 114, 121, 167
184	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	7	47, 48, 50, 52, 54, 83, 172
185	1	5	10	99, 100, 101, 102, 103, 104, 106, 143, 148, 167
186	1	9	10	141, 142, 143, 144, 149, 157, 160, 163, 166, 172
187	1	7	8	107, 108, 109, 110, 111, 112, 113, 114
188	1	9	10	67, 68, 69, 70, 76, 78, 80, 83, 124, 167
189	1	4	7	79, 80, 83, 85, 92, 95, 182
190	1	9	10	29, 30, 31, 32, 33, 34, 35, 36, 45, 168
191	3	1, 4, 5	10	3, 6, 9, 174, 184, 186, 191, 194, 197, 200
192	1	8	10	10, 11, 12, 13, 14, 15, 16, 17, 19, 168
193	1	8	10	37, 38, 39, 40, 41, 42, 43, 44, 54, 169
194	1	5	8	16, 17, 18, 19, 20, 21, 28, 177
195	1	4	10	66, 67, 68, 69, 70, 71, 72, 73, 82, 174
196	1	4	10	5, 6, 7, 8, 9, 10, 11, 12, 13, 183
197	2	2, 9	10	122, 123, 124, 125, 126, 127, 128, 130, 133, 168
198	1	4	8	115, 116, 118, 119, 137, 141, 145, 151
199	1	5	8	88, 89, 90, 91, 93, 94, 96, 168
200	1	2	7	16, 17, 18, 19, 20, 21, 169
201	3	3, 7, 9	8	135, 137, 138, 140, 142, 144, 146, 172
202	1	9	10	175, 176, 177, 178, 179, 180, 181, 182, 183, 184
203	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	70, 71, 72, 73, 74, 81, 83, 85, 87, 170
204	1	9	9	158, 159, 163, 165, 167, 171, 173, 175, 178
205	1	8	6	114, 115, 116, 117, 121, 122
206	1	7	10	57, 58, 59, 60, 61, 62, 63, 64, 83, 169
207	1	8	10	86, 87, 89, 91, 93, 95, 97, 99, 125, 169
208	1	4	10	3, 4, 5, 6, 7, 8, 9, 10, 12, 185
209	3	3, 7, 9	8	178, 179, 180, 181, 183, 184, 185, 186
210	3	3, 7, 9	6	96, 100, 108, 111, 114, 152
211	5	1, 4, 5, 7, 8	10	46, 47, 48, 49, 50, 51, 52, 53, 54, 169
212	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	1, 2, 3, 4, 5, 6, 7, 8, 178, 200
213	1	7	10	101, 103, 105, 107, 109, 111, 119, 122, 125, 170
214	3	3, 7, 9	9	26, 27, 28, 29, 30, 31, 32, 33, 34
215	1	5	8	96, 97, 98, 103, 105, 114, 121, 179
216	4	4, 5, 7, 8	10	24, 25, 26, 27, 28, 29, 30, 31, 41, 170
217	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	29, 30, 31, 32, 33, 34, 35, 37, 43, 171
218	3	3, 7, 9	10	154, 155, 156, 157, 158, 159, 160, 162, 165, 171
219	1	8	8	57, 59, 61, 63, 65, 68, 83, 180
220	1	4	10	118, 120, 123, 126, 129, 138, 142, 146, 150, 172

Fonte: Elaborado pelo autor.

Tabela 28: Banco de dados da Instância No. 3 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
221	1	5	10	105, 106, 107, 108, 109, 110, 111, 114, 121, 171
222	1	5	9	13, 17, 20, 23, 26, 35, 39, 43, 180
223	1	4	9	44, 46, 48, 50, 52, 54, 56, 58, 172
224	1	9	10	65, 67, 69, 71, 73, 75, 77, 79, 83, 172
225	2	2, 5	9	96, 98, 100, 102, 104, 112, 118, 122, 173
226	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	163, 164, 165, 166, 168, 169, 170, 171, 173
227	3	3, 7, 9	10	137, 138, 139, 140, 141, 142, 155, 158, 161, 175
228	3	3, 7, 9	10	146, 148, 150, 152, 154, 156, 165, 168, 171, 175
229	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	96, 97, 98, 99, 100, 102, 106, 115, 122, 173
230	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	32, 33, 34, 35, 36, 37, 38, 39, 43, 173
231	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	3	70, 72, 173
232	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	57, 58, 59, 60, 61, 63, 65, 68, 83, 173
233	1	2	10	115, 117, 123, 127, 131, 135, 139, 144, 149, 179
234	3	3, 7, 9	7	25, 26, 27, 37, 39, 51, 183
235	1	9	8	10, 11, 12, 13, 14, 15, 29, 174
236	1	9	10	122, 123, 124, 125, 126, 127, 128, 129, 130, 174
237	1	2	8	145, 146, 147, 151, 153, 163, 166, 175
238	3	1, 4, 5	10	66, 67, 68, 69, 70, 71, 72, 73, 83, 174
239	1	2	10	63, 64, 65, 66, 67, 68, 69, 70, 84, 174
240	1	4	10	26, 27, 28, 32, 34, 36, 38, 40, 43, 175
241	1	8	7	104, 106, 108, 110, 112, 115, 176
242	2	2, 9	10	29, 30, 31, 33, 35, 37, 39, 41, 44, 17
243	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	88, 89, 90, 91, 92, 93, 94, 95, 96, 181
244	3	1, 4, 5	10	125, 126, 127, 129, 131, 133, 135, 137, 139, 180
245	1	8	9	1, 4, 7, 12, 17, 192, 194, 196, 198
246	1	8	10	163, 164, 165, 166, 167, 168, 169, 170, 171, 176
247	1	2	10	1, 186, 187, 188, 189, 191, 193, 195, 197, 199
248	3	3, 7, 9	10	100, 101, 102, 103, 104, 105, 106, 115, 141, 177
249	1	2	9	35, 36, 37, 38, 39, 40, 41, 42, 44
250	3	3, 7, 9	10	47, 49, 51, 57, 60, 73, 78, 83, 88, 177
251	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	4	147, 148, 149, 151
252	1	5	10	98, 99, 100, 101, 102, 103, 112, 115, 140, 178
253	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	108, 109, 110, 111, 112, 119, 121, 131, 134
254	1	4	8	90, 91, 92, 93, 94, 95, 97, 98
255	5	1, 4, 5, 7, 8	9	2, 5, 8, 13, 20, 189, 190, 191, 199
256	3	1, 4, 5	10	49, 50, 51, 53, 55, 57, 59, 69, 84, 182
257	1	2	10	7, 8, 9, 10, 11, 12, 19, 37, 44, 178
258	1	8	10	2, 183, 184, 185, 190, 192, 194, 196, 198, 200
259	3	3, 7, 9	10	69, 70, 71, 72, 73, 74, 75, 76, 84, 182
260	1	2	9	99, 100, 102, 106, 108, 110, 120, 123, 179
261	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	123, 124, 128, 131, 134, 138, 142, 146, 153, 185
262	1	4	10	74, 75, 76, 77, 78, 79, 80, 81, 84, 179
263	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	106, 107, 108, 109, 110, 111, 113, 116, 167
264	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	5	75, 76, 86, 90, 96
265	3	3, 7, 9	9	113, 114, 115, 116, 117, 118, 119, 121, 180

Fonte: Elaborado pelo autor.

Tabela 28: Banco de dados da Instância No. 3 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
266	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	38, 40, 42, 48, 51, 57, 64, 85, 179
267	1	8	5	187, 188, 189, 190, 192
268	1	8	6	101, 104, 107, 137, 141, 180
269	1	9	9	1, 7, 13, 180, 184, 187, 189, 191, 193
270	3	1, 4, 5	10	6, 7, 8, 9, 10, 11, 12, 13, 14, 181
271	1	8	10	152, 153, 154, 155, 156, 157, 158, 159, 160, 180
272	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	94, 96, 98, 100, 103, 112, 116, 122, 194
273	1	2	8	155, 156, 159, 161, 163, 167, 169, 186
274	1	9	10	63, 64, 66, 68, 70, 74, 79, 84, 89, 181
275	1	9	9	170, 171, 172, 173, 175, 176, 177, 178, 181
276	3	3, 7, 9	8	2, 193, 194, 195, 196, 197, 198, 200
277	4	4, 5, 7, 8	6	101, 109, 112, 147, 154, 186
278	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	22, 23, 24, 29, 35, 39, 43, 47, 52, 181
279	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	2	180, 184
280	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	81, 82, 83, 84, 85, 86, 87, 88, 89, 181
281	1	5	9	56, 57, 58, 59, 60, 61, 62, 64, 85
282	1	5	10	127, 128, 129, 130, 131, 132, 133, 134, 135, 181
283	1	4	10	114, 115, 116, 117, 118, 119, 120, 121, 123, 185
284	1	8	10	35, 36, 37, 38, 39, 40, 41, 42, 44, 182
285	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	53, 56, 58, 60, 62, 64, 66, 77, 85, 183
286	1	8	10	145, 146, 147, 148, 155, 158, 161, 164, 168, 181
287	1	9	10	42, 43, 44, 45, 46, 47, 48, 49, 86, 184
288	3	3, 7, 9	4	15, 16, 17, 181
289	1	7	6	155, 156, 162, 170, 173, 182
290	1	2	6	43, 44, 46, 56, 65, 182
291	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	7	103, 107, 108, 110, 116, 122, 200
292	1	7	10	100, 101, 102, 103, 104, 105, 106, 122, 124, 186
293	1	9	9	94, 95, 96, 98, 100, 103, 112, 116, 123
294	1	8	9	57, 58, 59, 60, 61, 68, 70, 72, 95
295	1	5	7	176, 177, 178, 179, 180, 181, 187
296	1	7	1	182
297	1	9	9	66, 67, 68, 69, 70, 71, 72, 73, 86
298	1	5	10	170, 171, 172, 173, 174, 175, 177, 179, 181, 184
299	3	1, 4, 5	10	18, 20, 22, 24, 26, 28, 30, 32, 87, 185
300	1	7	9	3, 6, 9, 14, 21, 193, 195, 197, 200
301	1	8	10	1, 190, 191, 192, 193, 194, 195, 196, 197, 198
302	2	2, 9	9	36, 37, 41, 43, 45, 47, 49, 51, 182
303	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	62, 65, 67, 69, 71, 75, 80, 85, 90, 183
304	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 183
305	3	1, 4, 5	10	4, 5, 6, 7, 8, 9, 10, 11, 14, 183
306	1	4	1	152
307	1	8	7	2, 194, 195, 197, 198, 199, 200
308	1	7	5	77, 81, 85, 123, 195
309	1	7	8	98, 99, 100, 101, 103, 107, 119, 123
310	3	3, 7, 9	9	45, 47, 49, 51, 53, 55, 57, 91, 97

Fonte: Elaborado pelo autor.

Tabela 28: Banco de dados da Instância No. 3 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
311	1	2	10	92, 93, 94, 95, 101, 104, 107, 116, 123, 192
312	1	5	8	1, 5, 15, 180, 181, 182, 188, 191
313	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	8	24, 25, 26, 28, 29, 30, 31, 184
314	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	35, 36, 37, 38, 39, 40, 41, 42, 45, 193
315	1	4	5	101, 104, 107, 124, 186,
316	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	17, 19, 21, 25, 27, 30, 33, 45, 184
317	2	2, 9	10	138, 139, 140, 141, 142, 143, 144, 145, 146, 185
318	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	67, 68, 69, 70, 71, 73, 75, 78, 85, 185
319	1	4	10	3, 117, 119, 121, 123, 125, 127, 129, 138, 142
320	1	7	10	123, 124, 128, 130, 132, 134, 136, 138, 140, 185
321	1	4	9	98, 103, 105, 107, 109, 111, 122, 125, 193
322	3	1, 4, 5	8	60, 61, 62, 63, 64, 65, 85, 185
323	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	4, 61, 62, 63, 64, 65, 66, 67, 70
324	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	18, 19, 20, 21, 22, 23, 24, 26, 194
325	1	4	9	38, 39, 40, 41, 42, 43, 44, 45, 46
326	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	30, 31, 32, 33, 34, 47, 57, 74, 87
327	1	5	9	143, 144, 145, 146, 151, 153, 156, 159, 171
328	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	5	65, 71, 74, 83, 87
329	1	8	9	36, 37, 38, 39, 40, 43, 45, 86, 186
330	3	3, 7, 9	10	9, 12, 15, 186, 189, 190, 191, 192, 193, 194
331	1	4	6	124, 126, 134, 138, 142, 186
332	3	1, 4, 5	9	107, 108, 109, 110, 111, 112, 113, 117, 124
333	3	3, 7, 9	10	30, 31, 32, 33, 36, 40, 44, 48, 55, 187
334	1	2	9	74, 75, 76, 77, 78, 79, 80, 81, 86
335	1	4	8	10, 15, 22, 178, 179, 183, 185, 187
336	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	18, 19, 20, 21, 22, 23, 24, 25, 26, 187
337	1	9	9	46, 188, 189, 190, 192, 194, 196, 198, 200
338	1	5	9	157, 158, 159, 160, 161, 162, 163, 164, 165
339	1	5	8	32, 33, 34, 35, 42, 44, 46, 87
340	1	7	10	14, 15, 18, 21, 24, 27, 36, 40, 47, 187
341	3	1, 4, 5	10	98, 99, 101, 103, 105, 107, 109, 117, 125, 188
342	1	9	10	7, 8, 9, 10, 11, 12, 13, 14, 15, 194
343	5	1, 4, 5, 7, 8	9	87, 88, 89, 90, 93, 110, 118, 126, 189
344	1	7	9	172, 173, 174, 175, 176, 177, 178, 179, 188
345	1	9	10	138, 139, 140, 141, 142, 143, 144, 145, 146, 188
346	1	7	4	42, 43, 44, 55
347	1	7	6	25, 27, 29, 31, 35, 37
348	1	7	10	2, 182, 183, 184, 185, 186, 187, 188, 189, 199
349	1	8	9	87, 88, 89, 90, 91, 92, 93, 94, 95
350	1	2	10	28, 29, 30, 31, 32, 33, 37, 41, 48, 188
351	1	5	8	2, 15, 189, 190, 192, 194, 196, 200
352	1	8	9	17, 18, 19, 20, 21, 22, 23, 24, 25
353	1	8	7	49, 52, 53, 54, 69, 72, 125
354	1	2	10	7, 8, 9, 10, 11, 12, 13, 14, 16, 189
355	1	4	4	26, 28, 30, 36

Fonte: Elaborado pelo autor.

Tabela 28: Banco de dados da Instância No. 3 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
356	3	1, 4, 5	9	1, 47, 186, 187, 191, 193, 195, 197, 199
357	1	2	10	185, 186, 187, 188, 189, 190, 191, 192, 193, 195
358	3	1, 4, 5	10	179, 180, 181, 182, 184, 186, 188, 190, 192, 196
359	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	128, 129, 130, 131, 132, 133, 134, 135, 136, 197
360	1	9	5	36, 38, 40, 55, 190
361	3	3, 7, 9	9	154, 155, 156, 157, 158, 159, 160, 161, 162
362	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	160, 161, 162, 163, 164, 165, 166, 167, 168, 189
363	2	4, 5	9	97, 99, 101, 103, 105, 113, 116, 119, 132
364	1	5	10	106, 107, 108, 109, 110, 111, 113, 117, 125, 190
365	1	7	10	153, 154, 155, 156, 157, 158, 159, 160, 161, 198
366	3	1, 4, 5	4	107, 109, 113, 117
367	1	9	10	146, 147, 148, 149, 150, 151, 152, 153, 155, 195
368	1	7	9	6, 91, 93, 95, 97, 99, 108, 117, 132
369	2	4, 5	9	159, 160, 161, 163, 164, 165, 166, 167, 191
370	1	9	10	127, 128, 129, 130, 131, 132, 135, 139, 143, 191
371	1	9	10	133, 134, 135, 136, 137, 138, 139, 140, 141, 194
372	1	2	9	77, 78, 84, 91, 94, 111, 119, 127, 191
373	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	46, 47, 48, 49, 50, 51, 52, 53, 97, 192
374	1	8	10	28, 29, 30, 31, 32, 33, 34, 35, 86, 192
375	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	8	86, 93, 95, 97, 99, 117, 126, 192
376	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	112, 113, 114, 115, 116, 117, 120, 123, 126
377	1	5	10	5, 131, 132, 133, 134, 135, 136, 137, 139, 143
378	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	67, 68, 69, 70, 71, 72, 73, 74, 86
379	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	113, 114, 115, 116, 117, 118, 126, 128, 130, 192
380	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	4	132, 133, 134, 136
381	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	3, 13, 14, 15, 16, 17, 18, 27, 38, 87
382	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	145, 146, 147, 148, 149, 150, 151, 152, 156, 193
383	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	56, 57, 58, 59, 60, 61, 76, 79, 87, 199
384	1	5	9	31, 32, 33, 36, 38, 40, 42, 88, 196
385	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	6	6, 79, 80, 81, 82, 83
386	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	8	54, 55, 56, 57, 58, 59, 89, 197
387	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	77, 78, 79, 80, 81, 82, 83, 84, 88, 193
388	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	124, 125, 126, 127, 128, 129, 130, 131, 133
389	3	1, 4, 5	9	157, 158, 160, 162, 166, 168, 170, 182, 194
390	3	1, 4, 5	10	66, 68, 70, 72, 74, 76, 78, 80, 88, 194
391	3	1, 4, 5	10	189, 190, 191, 192, 193, 194, 195, 196, 197, 200
392	5	1, 4, 5, 7, 8	8	132, 133, 134, 135, 136, 137, 161, 195
393	5	1, 4, 5, 7, 8	6	92, 95, 104, 108, 120, 126
394	1	2	10	190, 191, 192, 193, 194, 195, 196, 197, 198, 199
395	1	2	9	47, 48, 52, 54, 56, 58, 60, 71, 89
396	3	3, 7, 9	7	22, 23, 24, 25, 26, 27, 196
397	1	2	9	147, 148, 149, 150, 152, 154, 157, 160, 172
398	1	2	8	71, 72, 73, 74, 75, 76, 89, 197
399	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10	50, 51, 52, 57, 59, 62, 65, 118, 127, 197
400	1	2	10	56, 57, 58, 59, 60, 61, 62, 74, 89, 198

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4.

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
1	6	1, 2, 4, 7, 9, 10	14	65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78,
2	3	2, 5, 6	18	33, 34, 35, 36, 37, 38, 39, 41, 43, 45, 47, 50, 54, 58, 64, 72, 79, 243
3	2	4, 8	1	57
4	1	2	18	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 53, 55, 58, 64, 72, 80, 242
5	1	2	16	2, 225, 226, 227, 228, 233, 235, 237, 239, 241, 245, 248, 265, 281, 285, 292
6	2	2, 8	1	241
7	1	5	17	154, 155, 156, 157, 158, 164, 166, 168, 170, 172, 174, 176, 178, 182, 196, 235, 242
8	1	8	1	293
9	3	2, 5, 6	18	101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 120, 125, 130
10	1	9	16	63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 80
11	6	2, 3, 4, 5, 6, 7	18	86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 111
12	3	2, 4, 7	18	86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 112
13	1	8	1	7
14	3	1, 2, 4	14	43, 44, 45, 46, 47, 53, 55, 57, 59, 61, 73, 76, 79, 82
15	5	1, 2, 3, 4, 5	7	134, 139, 148, 154, 156, 158, 160
16	1	5	12	31, 32, 33, 35, 37, 39, 41, 43, 64, 72, 81, 242
17	3	4, 7, 9	16	117, 118, 119, 122, 126, 129, 132, 135, 138, 141, 144, 158, 162, 167, 184, 223
18	1	8	1	24
19	5	2, 5, 6, 9, 10	12	144, 146, 147, 149, 153, 155, 157, 159, 197, 209, 231, 243
20	2	4, 7	17	48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 65, 82, 115
21	2	4, 5	17	117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 229
22	2	4, 8	1	93
23	1	2	18	3, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 289, 292
24	3	2, 3 10	14	92, 95, 103, 105, 107, 109, 111, 113, 115, 117, 119, 120, 121, 123
25	1	4	18	32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 55, 59, 65, 73, 103, 112
26	1	4	17	6, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 184, 210
27	2	2, 10	12	251, 253, 254, 256, 257, 259, 260, 261, 262, 263, 264, 265
28	1	5	11	7, 11, 14, 251, 252, 255, 271, 274, 280, 295, 299
29	4	2, 3, 4, 5	17	110, 112, 115, 118, 121, 124, 127, 130, 133, 154, 159, 164, 169, 174, 208, 213, 243
30	2	2, 10	14	252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 266

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
31	2	2, 3	18	20, 24, 27, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288
32	5	2, 5, 6, 9, 10	8	112, 274, 276, 277, 278, 280, 281, 282
33	6	2, 3, 4, 5, 6, 7	18	34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 55, 59, 65, 73, 97, 114
34	1	8	1	193
35	1	5	18	5, 191, 192, 193, 194, 195, 197, 199, 201, 203, 205, 207, 209, 211, 213, 215, 217, 223
36	1	9	16	267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 285
37	3	2, 5, 6	18	63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 74, 77, 80, 83, 85, 89, 98, 116
38	6	1, 2, 3, 4, 5, 8	1	125
39	1	2	18	28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 41, 59, 91, 93, 111, 116
40	2	2, 3	10	282, 283, 284, 285, 291, 293, 294, 295, 298, 300
41	1	2	17	100, 101, 105, 111, 114, 117, 120, 123, 126, 129, 132, 144, 152, 211, 230, 234, 243
42	5	1, 2, 3, 4, 5	14	249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262
43	1	2	8	83, 86, 87, 88, 89, 90, 95, 123
44	1	8	0	
45	3	2, 5, 6	18	43, 44, 45, 46, 48, 50, 58, 68, 73, 77, 81, 85, 89, 93, 98, 101, 104, 124
46	3	2, 5, 6	17	197, 198, 199, 200, 202, 203, 204, 205, 206, 207, 208, 209, 210, 227, 231, 235, 244
47	1	8	1	9
48	2	4, 5	18	3, 138, 139, 140, 141, 142, 143, 144, 147, 158, 162, 166, 170, 174, 178, 198, 217, 224
49	2	4, 5	18	4, 148, 149, 150, 151, 152, 153, 154, 155, 159, 163, 167, 171, 175, 179, 199, 218, 225
50	4	2, 3, 4, 5	16	6, 10, 14, 18, 22, 26, 30, 42, 59, 66, 73, 98, 243, 295, 296, 300
51	2	2, 3	18	142, 143, 144, 145, 146, 147, 148, 149, 151, 153, 155, 157, 159, 161, 198, 224, 227, 244
52	2	4, 7	18	4, 190, 191, 193, 194, 195, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 215, 232
53	2	2, 5	18	3, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 281, 285, 293
54	1	8	1	69
55	6	1, 2, 4, 7, 9, 10	10	187, 188, 189, 190, 191, 192, 193, 196, 197, 198
56	1	5	18	40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 55, 59, 66, 73, 99, 244
57	1	8	0	
58	3	4, 7, 9	16	167, 168, 169, 171, 173, 175, 177, 179, 181, 183, 185, 187, 189, 191, 223, 226

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
59	3	1, 2, 4	13	120, 121, 123, 126, 129, 153, 160, 166, 172, 178, 184, 190, 196
60	2	4, 7	18	147, 149, 151, 153, 159, 162, 165, 168, 171, 174, 177, 180, 183, 189, 199, 224, 228, 245
61	2	2, 4	18	185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 235, 238, 244
62	1	2	18	59, 60, 61, 62, 63, 64, 65, 70, 74, 78, 82, 86, 90, 94, 99, 102, 105, 129
63	2	2, 3	9	175, 185, 198, 201, 204, 208, 212, 224, 244
64	1	4	18	178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 199, 210, 232, 244
65	1	8	1	248
66	3	2, 5, 6	17	1, 5, 9, 14, 273, 274, 275, 276, 277, 278, 279, 283, 285, 287, 289, 293, 297
67	4	2, 3, 4, 5	16	81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 92, 97, 102, 108, 119, 124
68	2	2, 10	14	212, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226
69	2	2, 10	14	137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 152
70	2	4, 7	17	4, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 85
71	2	4, 7	17	87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 102, 103, 129
72	1	2	18	43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 72, 74, 99, 245
73	2	2, 3	16	8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 66, 113, 130
74	3	1, 2, 4	12	93, 94, 96, 98, 100, 104, 106, 108, 110, 114, 116, 118
75	1	4	18	1, 3, 5, 7, 245, 279, 280, 281, 282, 283, 285, 287, 289, 291, 293, 295, 297, 299
76	3	2, 5, 6	17	1, 7, 15, 222, 223, 224, 225, 226, 227, 228, 230, 231, 232, 233, 258, 261, 282
77	5	2, 5, 6, 9, 10	6	270, 281, 282, 283, 286, 293
78	1	4	9	176, 178, 182, 184, 186, 216, 218, 224, 245
79	2	2, 3	18	2, 6, 10, 15, 248, 249, 250, 251, 253, 255, 257, 259, 261, 263, 265, 290, 294, 298
80	2	2, 8	1	158
81	9	1, 2, 3, 4, 5, 6, 7, 9, 10	9	283, 284, 285, 286, 287, 288, 289, 291, 296
82	2	2, 3	18	3, 8, 13, 18, 23, 28, 81, 100, 131, 265, 266, 267, 268, 269, 270, 271, 272, 298
83	5	1, 2, 3, 4, 5	14	144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 181, 185

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
84	2	2, 10	14	107, 108, 109, 110, 115, 117, 119, 121, 123, 125, 127, 141, 145, 152
85	3	4, 5, 7	17	201, 202, 203, 204, 205, 206, 207, 209, 210, 211, 212, 213, 214, 215, 217, 237, 245
86	5	2, 5, 6, 9, 10	14	4, 132, 133, 134, 135, 136, 137, 138, 140, 141, 142, 168, 173, 199
87	1	5	15	2, 8, 16, 243, 244, 246, 247, 250, 251, 252, 254, 256, 259, 277, 283
88	1	5	7	202, 203, 204, 206, 208, 209, 226
89	1	8	1	258
90	1	8	1	267
91	1	9	16	247, 248, 249, 250, 251, 257, 259, 261, 263, 265, 267, 269, 271, 273, 282, 286
92	4	2, 3, 4, 5	16	215, 216, 217, 218, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 233, 245
93	4	2, 5, 6, 9	16	94, 95, 96, 97, 99, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126
94	1	2	15	165, 166, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 227
95	3	4, 5, 7	17	109, 113, 116, 119, 122, 125, 128, 131, 150, 155, 160, 165, 170, 202, 209, 214, 246
96	3	2, 5, 6	17	3, 7, 11, 15, 19, 23, 27, 31, 35, 39, 60, 67, 74, 100, 246, 294, 297
97	4	2, 5, 6, 9	16	81, 82, 83, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 108, 109, 246
98	1	2	17	161, 163, 165, 167, 169, 171, 173, 175, 185, 188, 191, 194, 197, 205, 211, 226, 246
99	1	2	5	97, 116, 121, 126, 247
100	1	2	15	1, 2, 4, 8, 12, 16, 20, 24, 28, 36, 55, 61, 76, 101, 247
101	3	2, 9, 10	11	5, 9, 13, 17, 21, 25, 29, 56, 62, 298, 299
102	1	2	17	1, 3, 5, 7, 9, 15, 17, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300
103	1	2	16	5, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 56, 60, 101
104	6	2, 3, 4, 5, 6, 7	18	156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 182, 186, 199, 211, 227, 246
105	4	4, 5, 7, 9	12	5, 270, 271, 272, 273, 274, 275, 276, 277, 282, 286, 293
106	4	2, 4, 7, 9	16	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 21, 300
107	2	4, 7	18	149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 198, 202, 206, 212, 227, 247
108	4	4, 7, 8, 9	1	276

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
109	3	2, 9, 10	2	11, 12
110	4	2, 3, 4, 5	18	48, 49, 50, 51, 52, 54, 56, 58, 60, 62, 75, 78, 81, 84, 86, 90, 102, 248
111	1	2	17	119, 122, 124, 127, 130, 148, 154, 161, 167, 173, 179, 185, 191, 197, 202, 211, 228
112	2	2, 5	18	2, 4, 6, 8, 10, 13, 16, 18, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290
113	1	9	16	164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 179, 200, 211
114	6	1, 2, 4, 7, 9, 10	14	79, 80, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106
115	3	1, 2, 4	14	247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 261
116	2	2, 3	5	6, 132, 153, 200, 293
117	3	1, 2, 4	14	21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 37, 95
118	2	4, 7	18	23, 66, 67, 68, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 102
119	3	2, 5, 6	18	6, 240, 241, 242, 243, 244, 246, 248, 250, 252, 254, 256, 258, 260, 262, 264, 287, 293
120	2	4, 7	18	3, 247, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 287
121	3	2, 9, 10	6	175, 177, 179, 181, 183, 185
122	9	1, 2, 3, 4, 5, 6, 7, 9, 10	14	92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105
123	5	1, 2, 3, 4, 5	14	22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 37, 60
124	2	2, 10	3	31, 33, 67
125	1	2	17	7, 236, 237, 238, 239, 245, 249, 251, 253, 255, 257, 259, 261, 263, 266, 288, 294
126	1	9	14	21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 56, 60, 67, 75
127	3	2, 9, 10	14	35, 36, 37, 38, 39, 41, 43, 45, 47, 49, 51, 53, 56, 60
128	5	2, 5, 6, 9, 10	14	22, 23, 24, 26, 50, 55, 60, 65, 70, 75, 80, 84, 102, 248
129	3	2, 5, 6	18	21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 35, 38, 70, 73, 75, 228, 248
130	2	4, 7	18	101, 102, 107, 110, 113, 116, 119, 122, 125, 128, 131, 134, 152, 156, 200, 212, 229, 249
131	1	9	16	63, 64, 65, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94
132	3	4, 7, 9	13	8, 9, 10, 11, 16, 18, 20, 30, 39, 91, 96, 103, 248
133	1	5	17	15, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 56, 60, 83, 103
134	1	2	18	15, 250, 251, 252, 253, 254, 255, 256, 257, 266, 268, 270, 272, 274, 277, 282, 288, 294
135	2	2, 10	14	185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
136	2	2, 3	10	10, 11, 12, 13, 19, 22, 25, 28, 103, 249
137	1	5	18	4, 9, 14, 19, 24, 29, 82, 104, 249, 274, 275, 276, 277, 278, 284, 286, 295, 299
138	2	2, 3	17	117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 154, 236, 239, 249
139	3	2, 9, 10	10	130, 137, 138, 139, 140, 141, 147, 157, 200, 224
140	4	2, 5, 6, 9	16	185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 197, 198, 200, 212, 229, 251
141	2	4, 7	18	132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 147, 157, 228, 233, 238, 249
142	6	1, 2, 4, 7, 9, 10	4	183, 184, 198, 203
143	3	2, 5, 6	18	99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 250
144	3	1, 2, 4	14	229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242
145	3	2, 9, 10	8	46, 47, 48, 50, 51, 54, 56, 58
146	4	4, 5, 7, 9	16	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 296, 297, 298, 299, 300
147	1	8	0	
148	2	2, 10	14	44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 71,
149	2	2, 8	1	213
150	1	9	16	127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 141, 157, 159, 201, 213
151	3	1, 2, 4	14	1, 2, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300
152	3	1, 2, 4	6	9, 16, 284, 291, 293, 295
153	3	4, 7, 9	16	129, 130, 131, 132, 134, 136, 145, 156, 160, 164, 168, 172
154	3	2, 5, 6	16	176, 180, 215, 219
155	3	1, 2, 4	18	218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 235, 250
156	3	1, 2, 4	14	171, 172, 173, 174, 175, 176, 177, 178, 179, 181, 182, 183, 184, 190
156	1	2	18	16, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 142, 184, 191, 201, 225, 229
157	5	1, 2, 3, 4, 5	14	5, 10, 15, 20, 25, 279, 280, 281, 282, 283, 285, 287, 296, 300
158	2	2, 5	17	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 296, 297, 298, 299, 300
159	3	2, 5, 6	18	111, 114, 117, 120, 123, 126, 129, 132, 151, 156, 161, 166, 171, 204, 210, 215, 230, 250
160	3	4, 7, 9	13	160, 162, 181, 185, 189, 193, 197, 213, 218, 223, 228, 247 252
161	1	8	1	124
162	3	2, 3, 8	1	143
163	1	4	15	87, 88, 89, 90, 95, 99, 101, 103, 105, 107, 109, 111, 113, 115, 250
164	1	8	1	252

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
165	2	2, 5	18	145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 201, 213, 231, 256
166	3	1, 2, 4	8	228, 229, 230, 231, 232, 233, 234, 235
167	2	4, 7	17	161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 202, 214, 231
168	1	2	18	24, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114
169	6	2, 3, 4, 5, 6, 7	18	106, 107, 108, 109, 112, 115, 118, 121, 124, 127, 130, 133, 148, 158, 228, 232, 236, 251
170	1	2	18	184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 201, 213, 231, 253
171	4	2, 5, 6, 9	16	83, 84, 85, 89, 91, 93, 95, 97, 101, 103, 105, 107, 109, 111, 112, 114
172	2	2, 10	14	8, 9, 12, 14, 16, 18, 20, 22, 24, 26, 38, 41, 56, 73
173	1	2	15	176, 177, 178, 179, 180, 181, 182, 193, 199, 201, 205, 211, 216, 232, 253
174	1	4	17	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 68, 75, 237, 254
175	2	2, 10	14	134, 135, 136, 137, 142, 144, 146, 148, 152, 157, 162, 167, 172, 206
176	3	1, 2, 4	14	133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 145, 148, 159
177	3	2, 5, 6	17	153, 154, 155, 156, 157, 158, 178, 186, 190, 194, 198, 214, 219, 224, 243, 248, 255
178	1	8	1	188
179	3	1, 2, 4	9	83, 84, 85, 86, 87, 88, 89, 90, 91
180	2	2, 5	8	186, 199, 202, 206, 209, 213, 232, 257
181	1	2	17	44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 57, 74, 78, 82, 104
182	6	1, 2, 3, 4, 5, 8	1	221
183	9	1, 2, 3, 4, 5, 6, 7, 9, 10	13	116, 120, 123, 127, 130, 133, 136, 139, 142, 145, 159, 163, 168
184	1	5	14	17, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 104
185	5	2, 5, 6, 9, 10	7	179, 180, 181, 182, 183, 202, 213
186	3	1, 2, 4	14	104, 105, 108, 111, 114, 117, 120, 123, 126, 129, 132, 149, 153, 159
187	1	2	18	19, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 216, 220, 300
188	2	2, 3	17	19, 239, 240, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255
189	1	5	12	20, 73, 74, 75, 76, 77, 79, 81, 82, 92, 109, 112
190	4	2, 3, 4, 5	17	20, 235, 236, 237, 238, 239, 240, 241, 242, 243, 245, 246, 260, 262, 266, 289, 294

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
191	2	2, 3	16	129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 142, 148, 160, 202, 214
192	2	2, 10	13	10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 23
193	1	8	0	
194	1	5	17	27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 39, 42, 70, 75, 79, 83, 105
195	2	2, 3	16	25, 132, 135, 138, 144, 155, 162, 168, 174, 180, 186, 192, 198, 203, 214, 233
196	1	9	6	238, 239, 240, 242, 243, 246
197	7	2, 3, 4, 5, 6 7, 9	15	25, 91, 92, 93, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 116
198	1	8	1	76
199	5	2, 5, 6, 9, 10	14	40, 41, 42, 47, 49, 56, 66, 71, 75, 79, 83, 87, 91, 96
200	3	4, 7, 9	14	151, 152, 153, 154, 155, 156, 157, 158, 159, 170, 173, 176, 202, 214
201	4	4, 5, 7, 9	11	4, 8, 12, 26, 247, 248, 249, 250, 272, 296, 300
202	2	2, 10	14	4, 5, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30
203	4	2, 3, 4, 5	16	26, 137, 138, 139, 140, 141, 142, 144, 145, 146, 147, 149, 160, 217, 219, 233
204	1	5	17	27, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 192, 214, 234
205	2	2, 10	7	25, 27, 28, 29, 62, 76, 116
206	3	1, 2, 4	10	204, 206, 207, 208, 209, 210, 211, 212, 213, 215
207	1	4	12	44, 45, 46, 47, 52, 53, 54, 56, 61, 69, 77, 115
208	6	1, 2, 4, 7, 9, 10	14	187, 188, 190, 193, 196, 199, 202, 205, 208, 211, 214, 217, 220, 223
209	6	1, 2, 4, 7, 9, 10	13	14, 15, 16, 17, 18, 51, 61, 66, 71, 76, 81, 85, 104
210	1	2	17	6, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 61, 69, 77, 104
211	2	2, 10	14	134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 154, 156, 158, 160
212	3	1, 2, 4	14	252, 253, 254, 255, 256, 258, 260, 262, 264, 266, 268, 270, 272, 278
213	3	1, 2, 4	12	58, 59, 60, 61, 62, 64, 65, 66, 67, 68, 69, 71
214	1	5	8	32, 34, 36, 40, 54, 59, 64, 105
215	1	5	17	177, 178, 179, 180, 181, 182, 183, 184, 186, 189, 192, 195, 199, 203, 207, 215, 234
216	3	2, 5, 6	17	224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240
217	4	2, 3, 4, 5	15	145, 146, 147, 149, 150, 151, 153, 154, 155, 156, 158, 180, 203, 215, 234
218	2	4, 5	14	121, 125, 128, 131, 134, 137, 140, 143, 146, 160, 164, 169, 215, 234
219	3	1, 2, 4	14	50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63
220	2	4, 5	14	1, 2, 3, 6, 8, 9, 10, 11, 12, 295, 296, 297, 298, 299

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
221	1	4	16	141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 179, 183, 203, 236
222	3	4, 7, 9	6	164, 165, 168, 203, 216, 237
223	1	2	11	7, 8, 9, 14, 20, 23, 26, 29, 55, 65, 106
224	3	2, 9, 10	14	30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 61
225	2	2, 3	17	3, 227, 228, 229, 230, 231, 232, 233, 234, 235, 245, 247, 249, 251, 253, 255, 256
226	2	2, 4 2	17	135, 136, 137, 138, 139, 140, 141, 150, 152, 154, 156, 158, 160, 162, 204, 225, 235
227	2	2, 8	0	
228	2	2, 3	17	1, 230, 231, 232, 233, 234, 235, 242, 244, 246, 248, 250, 252, 254, 256, 258, 260
229	1	4	17	169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 204, 216, 248
230	1	8	1	5
231	2	2, 10	7	265, 266, 267, 268, 269, 283, 285
232	2	2, 3	16	31, 32, 33, 34, 40, 42, 44, 46, 48, 50, 52, 54, 57, 62, 77, 110
233	2	2, 3	17	6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22
234	2	4, 7	17	1, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 297, 299
235	1	5	17	216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 235
236	3	4, 7, 9	9	143, 145, 150, 151, 152, 169, 193, 204, 295
237	3	4, 7, 9	16	184, 186, 191, 194, 197, 200, 203, 206, 209, 212, 215, 218, 221, 224, 229, 234
238	2	2, 10	13	27, 28, 29, 30, 31, 32, 42, 44, 46, 48, 51, 56, 62
239	1	2	17	1, 6, 11, 16, 21, 26, 78, 84, 105, 258, 259, 260, 261, 262, 263, 264, 273
240	3	1, 2, 4	13	229, 230, 231, 234, 236, 238, 240, 242, 244, 246, 249, 251, 271
241	4	2, 5, 6, 9	15	45, 47, 49, 51, 54, 57, 60, 70, 74, 78, 93, 98, 109, 120, 125
242	1	5	17	138, 139, 140, 141, 143, 145, 147, 149, 153, 158, 163, 168, 173, 207, 212, 217, 235
243	3	1, 2, 4	14	216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229
244	4	2, 5, 6, 9	15	14, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 272
245	1	2	16	125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 139, 153, 180, 184, 204, 237
246	3	4, 7, 9	16	10, 11, 13, 15, 17, 19, 21, 23, 25, 37, 40, 43, 71, 76, 80, 84
247	3	4, 7, 9	16	51, 52, 53, 54, 55, 57, 67, 72, 76, 80, 84, 88, 92, 97, 100, 103
248	3	2, 3, 10	10	181, 182, 183, 184, 185, 186, 194, 204, 218, 295
249	1	4	15	143, 144, 145, 146, 147, 148, 149, 150, 168, 171, 174, 177, 204, 218, 241
250	5	2, 5, 6, 9, 10	14	251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 271, 273, 275, 278

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
251	3	4, 5, 7	17	261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 272, 274, 276, 279, 297, 298, 299
252	2	2, 10	10	62, 63, 65, 66, 67, 68, 69, 70, 71, 72
253	3	2, 5, 6	17	154, 155, 156, 157, 160, 163, 166, 169, 172, 175, 178, 181, 185, 192, 205, 226, 236
254	2	2, 10	14	1, 287, 288, 289, 290, 291, 292, 294, 295, 296, 297, 298, 299, 300
255	1	9	16	86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101
256	3	1, 2, 4	5	29, 34, 36, 38, 40
257	1	8	1	180
258	3	1, 2, 4	7	20, 21, 40, 52, 57, 67, 77
259	1	8	1	270
260	2	2, 3	17	13, 258, 259, 260, 261, 262, 263, 264, 265, 267, 269, 271, 273, 275, 278, 283, 290
261	2	4, 5	17	2, 7, 12, 17, 22, 27, 79, 85, 106, 288, 289, 290, 291, 292, 293, 294, 297
262	1	8	1	275
263	2	2, 3	16	159, 160, 161, 162, 163, 165, 167, 169, 171, 173, 175, 177, 181, 193, 205, 233
264	1	9	6	144, 146, 149, 161, 210, 218
265	3	4, 7, 9	16	254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 279, 283, 290, 294
266	3	1, 2, 4	14	21, 22, 23, 24, 26, 28, 30, 32, 34, 36, 38, 40, 57, 62
267	2	2, 3	17	80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 105
268	2	4, 7	17	160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 181, 205, 234, 236
269	1	5	17	187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 205, 218, 237
270	2	2, 3	10	3, 4, 5, 6, 7, 35, 42, 92, 97, 106
271	2	2, 10	14	210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223
272	1	4	17	1, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 275, 279, 297, 299
273	3	4, 7, 9	16	3, 11, 286, 287, 288, 289, 290, 292, 293, 294, 295, 296, 297, 298, 299, 300
274	9	1, 2, 3, 4, 5, 6, 7, 9, 10	13	46, 47, 48, 50, 52, 54, 56, 58, 60, 62, 93, 98, 107
275	2	2, 10	12	86, 87, 88, 90, 92, 94, 96, 100, 102, 104, 108, 110
276	1	5	17	117, 118, 125, 128, 131, 150, 156, 163, 169, 175, 181, 187, 193, 199, 205, 219, 237
277	3	1, 2, 4	12	32, 33, 34, 42, 44, 46, 48, 50, 52, 54, 57, 63
278	4	2, 3, 4, 5	17	269, 270, 271, 272, 273, 274, 275, 277, 278, 279, 281, 283, 285, 287, 289, 291, 295
279	2	2, 10	14	115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128
280	2	2, 3	17	158, 160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 225, 230, 235, 239

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
281	3	2, 5, 6	17	209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 225, 238
282	1	5	7	194, 195, 196, 197, 200, 222, 228
283	3	2, 5, 6	17	11, 13, 15, 17, 19, 22, 25, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299
284	1	4	14	109, 110, 111, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 124
285	2	2, 3	17	83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 106
286	3	4, 7, 9	16	230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245
287	1	2	17	232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 249
288	1	8	1	71
289	2	2, 10	11	160, 161, 162, 163, 164, 165, 166, 167, 169, 172, 175
290	2	2, 3 2	15	111, 112, 113, 114, 116, 118, 120, 122, 124, 126, 128, 142, 150, 161, 219
291	3	1, 2, 4	11	269, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280
292	1	8	1	217
293	3	4, 7, 9	15	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 63, 70, 78
294	6	1, 2, 4, 7, 9, 10	12	44, 45, 49, 51, 53, 55, 57, 59, 61, 94, 99, 108
295	2	4, 7	17	2, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 298, 300
296	2	2, 10	14	118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131
297	1	8	1	192
298	3	1, 2, 4	14	221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234
299	4	2, 5, 6, 9	15	244, 245, 246, 247, 248, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259
300	1	4	15	133, 136, 139, 142, 151, 157, 164, 170, 176, 182, 188, 194, 200, 206, 220
301	2	2, 3	17	133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 179, 183, 194, 206, 219, 239
302	9	1, 2, 3, 4, 5, 6, 7, 9, 10	5	6, 7, 8, 9, 19
303	1	9	15	53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 66, 69, 72
304	1	8	1	52
305	2	2, 8	1	74
306	2	4, 5	17	138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 150, 154, 162, 206, 219, 239
307	1	4	17	168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 180, 184, 195, 207, 220, 240
308	4	2, 3, 4, 5	17	42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 54, 57, 63, 70, 78, 106
309	1	5	15	37, 38, 40, 41, 43, 45, 47, 49, 51, 53, 55, 58, 77, 79, 81
310	6	2, 3, 4, 5, 6, 7	17 17	74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 91, 107 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 91, 107

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
311	2	4, 5	16	28, 29, 30, 31, 32, 33, 34, 35, 36, 46, 48, 52, 67, 70, 73, 107
312	1	8	1	103
313	1	8	0	
314	1	9	14	29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 65, 79
315	2	2, 3	12	159, 163, 179, 183, 187, 191, 195, 210, 215, 225, 244, 249
316	3	2, 5, 6	17	236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 263, 288, 290, 296
317	3	1, 2, 4	7	270, 271, 272, 273, 284, 286, 288
318	4	2, 5, 6, 9	16	37, 38, 39, 40, 41, 42, 43, 44, 45, 47, 49, 65, 68, 71, 74, 108
319	4	2, 5, 6, 9	16	12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28
320	2	2, 10	5	144, 145, 146, 147, 148
321	2	4, 7	17	129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 140, 143, 151, 162, 206, 220, 239
322	2	2, 3	16	24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 98, 117, 122, 127
323	4	2, 3, 4, 5	15	164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 177, 182, 220
324	3	2, 4, 7	17	94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 110, 113
325	2	2, 5	12	86, 87, 88, 89, 90, 91, 93, 95, 99, 118, 123, 128
326	2	2, 3	16	64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 78, 80, 82
327	3	2, 8, 10	1	40
328	2	2, 5	15	206, 207, 208, 209, 210, 212, 213, 215, 216, 217, 218, 219, 220, 237, 240
329	3	4, 7, 9	16	102, 104, 110, 113, 116, 119, 122, 125, 128, 131, 143, 150, 163, 229, 233, 239
330	2	2, 10	14	113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126
331	3	1, 2, 4	14	286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299
332	4	2, 5, 6, 9	16	209, 210, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 226, 240
333	1	2	11	1, 5, 9, 13, 266, 267, 268, 269, 270, 273, 297
334	2	4, 5	7	92, 94, 96, 100, 119, 124, 129
335	3	4, 7, 9	11	161, 162, 163, 164, 165, 166, 167, 170, 194, 207, 300
336	1	8	1	205
337	4	2, 4, 7, 9	16	199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 211, 216, 221, 245, 250
338	2	4, 7	17	3, 7, 13, 245, 246, 247, 252, 254, 256, 258, 260, 262, 264, 266, 291, 295, 299
339	2	2, 3	17	144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 155, 157, 159, 163, 207, 221, 250
340	2	2, 10,	6	12, 13, 14, 15, 17, 36,

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
341	4	2, 3, 4, 5	17	46, 48, 50, 53, 55, 58, 61, 71, 75, 79, 94, 99, 104, 104, 110, 116, 121, 126
342	1	2	17	2, 4, 6, 107, 275, 276, 277, 278, 284, 286, 288, 290, 292, 294, 296, 298, 300
343	3	4, 7, 9	15	134, 140, 143, 152, 159, 165, 171, 177, 183, 189, 195, 201, 207, 221, 240
344	2	2, 5	16	76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 92
345	2	5, 9	8	226, 227, 228, 229, 230, 231, 232, 234
346	2	2, 10	10	131, 132, 133, 134, 135, 136, 140, 144, 150, 163
347	1	8	1	247
348	1	4	9	78, 79, 80, 81, 82, 83, 84, 85, 86
349	9	1, 2, 3, 4, 5, 6, 7, 9, 10	14	102, 103, 104, 105, 107, 109, 111, 113, 115, 117, 119, 121, 123, 125
350	1	9	16	162, 164, 166, 168, 170, 172, 174, 176, 187, 190, 193, 196, 200, 204, 208, 221
351	2	4, 7	17	58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 72, 77, 81, 85, 108
352	2	2, 3	8	35, 37, 38, 39, 105, 111, 122, 127
353	2	4, 7	17	1, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295
354	6	1, 2, 4, 7, 9, 10	9 9	49, 53, 63, 68, 73, 78, 82, 95, 109
355	1	4	17	2, 4, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 280, 284
356	2	2, 3	16	220, 221, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 240
357	1	9	16	260, 261, 262, 263, 264, 265, 266, 267, 268, 280, 282, 284, 286, 288, 290, 292
358	1	9	16	18, 19, 20, 25, 27, 29, 31, 33, 35, 37, 39, 53, 58, 63, 84, 96
359	2	2, 10	14	41, 42, 43, 44, 56, 59, 62, 72, 76, 80, 95, 100, 106, 112
360	5	1, 2, 3, 4, 5	14	107, 108, 109, 111, 113, 115, 123, 125, 127, 129, 131, 133, 232, 240
361	2	4, 5	17	148, 150, 152, 158, 161, 164, 167, 170, 173, 176, 179, 182, 186, 195, 207, 227, 241
362	2	2, 10	10	131, 132, 133, 134, 135, 136, 140, 144, 151, 163
363	1	2	13	4, 13, 262, 263, 264, 265, 266, 267, 268, 269, 271, 272, 273
364	2	2, 3	17	186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 199, 201, 208, 233, 241
365	2	4, 5	13	161, 176, 180, 184, 188, 192, 196, 212, 217, 222, 227, 246, 251
366	1	2	11	43, 44, 45, 46, 47, 49, 53, 58, 64, 71, 96
367	4	4, 5, 7, 9	15	147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 161, 166, 183, 221
368	2	2, 3	16	103, 106, 109, 112, 115, 118, 121, 124, 127, 130, 133, 151, 155, 164, 222, 241
369	3	4, 7, 9	7	2, 113, 114, 115, 117, 208, 222
370	1	4	14	217, 218, 219, 220, 222, 224, 225, 226, 227, 228, 229, 230, 231, 242

Fonte: Elaborado pelo autor.

Tabela 29: Banco de dados da Instância No. 4 (continuação).

Evento	No. de salas	Numeração das salas	No. de alunos	Denominação dos Alunos
371	3	4, 7, 9	15	63, 64, 65, 66, 67, 68, 69, 73, 77, 91, 101, 107, 118, 123, 128
372	2	4, 7	17	23, 26, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 289
373	1	4	16	135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 151, 208, 222, 241
374	1	4	12	15, 16, 17, 18, 21, 24, 27, 30, 61, 66, 97, 110
375	2	4, 7	17	77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 97, 242
376	2	2, 10	14	109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122
377	2	2, 5	11	10, 11, 12, 13, 19, 59, 69, 74, 83, 97, 111
378	1	9	16	187, 188, 189, 190, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 222
379	1	5	16	185, 189, 192, 195, 198, 201, 204, 207, 210, 213, 216, 219, 222, 226, 236, 241
380	2	4, 7	17	104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 120, 122
381	3	2, 5, 6	17	6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23
382	2	5, 9	14	175, 176, 177, 178, 179, 180, 181, 182, 185, 198, 200, 202, 209, 235
383	5	2, 5, 6, 9, 10	13	105, 106, 110, 112, 114, 124, 126, 128, 130, 132, 210, 236, 243
384	3	4, 7, 9	15	2, 6, 10, 282, 283, 284, 286, 287, 288, 289, 290, 291, 292, 294, 298
385	1	9	16	155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 196, 216, 236
386	2	2, 10	14	185, 186, 187, 188, 189, 197, 199, 201, 203, 205, 207, 209, 211, 213
387	2	4, 7	17	2, 236, 237, 238, 239, 240, 241, 243, 245, 247, 249, 251, 253, 255, 257, 259, 261
388	2	4, 7	17	187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 200, 203, 207, 211, 223, 242
389	3	4, 7, 9	16	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
390	3	2, 4, 7	17	159, 161, 163, 165, 167, 169, 171, 173, 175, 177, 179, 181, 183, 227, 232, 237, 242
391	6	1, 2, 4, 7, 9, 10	14	252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265
392	2	2, 10	14	126, 127, 128, 133, 135, 137, 146, 157, 161, 165, 169, 173, 177, 183
393	1	9	13	5, 234, 235, 236, 237, 238, 240, 253, 255, 257, 260, 281, 285
394	2	2, 9	16	263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 276, 280, 284, 291
395	1	9	16	274, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 291
396	2	5, 9	6	278, 279, 280, 290, 292, 294
397	1	9	16	195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 230
398	4	4, 5, 7, 9	16	257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 270, 281, 284, 291
399	1	9	16	4, 8, 267, 268, 269, 270, 271, 272, 280, 282, 284, 286, 288, 292, 296, 300
400	3	1, 2, 4	14	236, 237, 238, 239, 240, 241, 242, 243, 244, 246, 248, 250, 252, 254

Fonte: Elaborado pelo autor.