

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

RUBENS GODOFREDO SOARES CORREIA

FORMULAÇÕES MATEMÁTICAS E ALGORITMO HÍBRIDO PARA O
PROBLEMA DE PROGRAMAÇÃO DE SESSÕES TÉCNICAS DE
CONFERÊNCIAS

João Pessoa

2019

RUBENS GODOFREDO SOARES CORREIA

FORMULAÇÕES MATEMÁTICAS E ALGORITMO HÍBRIDO PARA O
PROBLEMA DE PROGRAMAÇÃO DE SESSÕES TÉCNICAS DE
CONFERÊNCIAS

Dissertação submetida ao Programa de Pós-Graduação em Informática do Centro de Informática da Universidade Federal da Paraíba, como requisito parcial para obtenção do Grau de Mestre em Informática.

Orientador: Prof. Dr. Anand Subramanian

Coorientador: Prof. Dr. Teobaldo Bulhões

João Pessoa

2019

Catálogo na publicação
Seção de Catalogação e Classificação

C824f Correia, Rubens Godofredo Soares.

FORMULAÇÕES MATEMÁTICAS E ALGORITMO HÍBRIDO PARA O PROBLEMA DE PROGRAMAÇÃO DE SESSÕES TÉCNICAS DE CONFERÊNCIAS / Rubens Godofredo Soares Correia. - João Pessoa, 2019.

72 f. : il.

Orientação: Anand Subramanian.

Coorientação: Teobaldo Bulhões.

Dissertação (Mestrado) - UFPB/Informática.

1. Otimização Combinatória. 2. Conference Scheduling.
3. Algoritmo híbrido. 4. Iterated local search. I.
Subramanian, Anand. II. Bulhões, Teobaldo. III. Título.

UFPB/BC



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Rubens Godofredo Soares Correia, candidato ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 13 de dezembro de 2019.

1 Aos treze dias do mês de dezembro, do ano de dois mil e dezenove, às dezesseis horas, no
2 Centro de Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se os
3 membros da Banca Examinadora constituída para julgar o Trabalho Final do Sr. Rubens
4 Godofredo Soares Correia, vinculado a esta Universidade sob a matrícula nº 20181000573,
5 candidato ao grau de Mestre em Informática, na área de "Sistemas de Computação", na linha
6 de pesquisa "Computação Distribuída", do Programa de Pós-Graduação em Informática, da
7 Universidade Federal da Paraíba. A comissão examinadora foi composta pelos professores:
8 Anand Subramanian (PPGI-UFPB) Orientador e Presidente da Banca, Lucídio dos Anjos
9 Formiga Cabral (PPGI-UFPB), Examinador Interno, Teobaldo Leite Bulhões Junior (UFPB),
10 Examinador Externo ao Programa, Puca Huachi Vaz Penna (UFOP), Examinador Externo à
11 Instituição. Dando início aos trabalhos, o Presidente da Banca cumprimentou os presentes,
12 comunicou aos mesmos a finalidade da reunião e passou a palavra ao candidato para que o
13 mesmo fizesse a exposição oral do trabalho de dissertação intitulado: "Formulações
14 Matemáticas e Algoritmo Híbrido para o Problema de Programação de Sessões Técnicas de
15 Conferências". Concluída a exposição, o candidato foi arguido pela Banca Examinadora que
16 emitiu o seguinte parecer: "**aprovado**". Do ocorrido, eu, Ruy Alberto Pisani Altafim,
17 Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai
18 assinada por mim e pelos membros da banca examinadora. João Pessoa, 13 de dezembro
19 de 2019.


Prof. Dr. Ruy Alberto Pisani Altafim

Prof. Dr. Anand Subramanian
Orientador (PPGI-UFPB)



Prof. Dr. Lucídio dos Anjos Formiga Cabral
Examinador Interno (PPGI-UFPB)



Prof. Dr. Teobaldo Leite Bulhões Junior
Examinador Externo ao Programa (UFPB)



Prof. Dr. Puca Huachi Vaz Penna
Examinador Externo à Instituição (UFOP)



AGRADECIMENTOS

- À minha esposa Ana Paula, pelo companheirismo, paciência e suporte durante todo o mestrado.
- À minha mãe, pelo seu apoio e presença em todos os momentos de minha vida, e a o meu pai que sempre aconselhou aos seus filhos estudarem.
- Ao meu irmão Carlos André, por sempre ter me incentivado a estudar e por ter me motivado a entrar no mestrado.
- Ao Prof. Anand Subramanian, pela excelente orientação, por toda sua ajuda, compreensão e confiança.
- Ao Prof. Teobaldo Bulhões, por todo o suporte que me deu sempre que precisei.
- Ao Prof. Puca Huachi, pela ajuda em tornar esse trabalho possível.
- Ao amigo Eduardo Queiroga, pelo apoio e incentivo.
- Aos companheiros do LASER pelo suporte e momentos de descontração proporcionados.
- Aos membros da banca da qualificação e da defesa, pelas correções e dicas.

RESUMO

Este trabalho trata do problema de programação de sessões técnicas de conferências. O objetivo é maximizar o benefício de alocar trabalhos com temas em comum em uma mesma sessão, satisfazendo restrições de recurso, bem como aquelas impostas pelos organizadores. Duas formulações matemáticas baseadas em programação linear inteira são apresentadas para o problema, que por sua vez é mostrado ser NP-Difícil. Dado o limite de escalabilidade dos modelos, um algoritmo híbrido que combina uma meta-heurística baseada em busca local e um procedimento baseado em programação matemática é proposto para resolver instâncias realísticas de grande porte. Experimentos computacionais foram conduzidos em instâncias de pequeno e médio porte, nas quais a heurística desenvolvida foi capaz de encontrar a maioria dos ótimos conhecidos. O algoritmo foi comparado com soluções geradas manualmente além de ter sido utilizado como ferramenta na geração da programação de algumas conferências realizadas no Brasil.

Palavras-chave: Otimização Combinatória. *Conference Scheduling*. Algoritmo híbrido. *Iterated local search*.

ABSTRACT

This work addresses the conference scheduling problem of technical sessions. The objective is to maximize the benefit of scheduling papers with common topics in the same session, satisfying resources constraints, as well as those imposed by the organizers. Two mathematical formulations based on integer linear programming are presented for the problem, which in turn is NP-Hard. Given the limit of the models' scalability, a hybrid algorithm that combines a metaheuristic based on local search and a procedure based on mathematical programming is proposed to solve realistic large size instances. Computational experiments were conducted in small and medium size instances, where the developed heuristic was capable of finding most known optima. The algorithm was compared with manually generated solutions, and it was also used as a tool to generate the schedule of some conferences in Brazil.

Keywords: Combinatorial Optimization. Conference Scheduling. Hybrid Algorithm. Iterated local search.

SUMÁRIO

Abstract	vi
1 Introdução	1
1.1 Definição do tema	1
1.2 Justificativa	2
1.3 Objetivos	4
1.3.1 Objetivo geral	4
1.3.2 Objetivos específicos	4
1.4 Estrutura do trabalho	4
2 Trabalhos relacionados	6
3 Definição do problema e formulação matemática	13
3.1 Definição do Problema	14
3.2 Formulações Matemáticas	17
3.2.1 Formulação F1	17
3.2.2 Formulação F2	18
4 Algoritmos propostos	22
4.1 Branch-and-cut com separação <i>lazy</i>	22
4.1.1 Problema de separação	22
4.1.2 Limites primais	24
4.2 O algoritmo híbrido HILS	24
4.2.1 Abordagem baseada em particionamento de conjuntos	27
4.2.2 Geração da solução inicial	28
4.2.3 Busca local	28

4.2.4	Estruturas de Vizinhança	29
4.2.5	Estruturas de Dados Auxiliares (EDAs)	30
4.2.6	Mecanismos de perturbação	31
5	Resultados	32
5.1	Calibração do algoritmo híbrido	32
5.2	Resultados obtidos em instâncias de pequeno porte	33
5.2.1	Resultados obtidos em instâncias de grande porte	41
5.2.2	Resultados obtidos em eventos reais	49
5.2.3	Aplicações	50
6	Conclusão	52
	Apêndice A Imagens das aplicações web	57

LISTA DE FIGURAS

A.1	Tela com exibição da solução com possibilidade de edição	57
A.2	Exibição dos temas das sessões por faixa de horário	58
A.3	Exibição de todas as sessões de uma faixa de horário e choques de auto- res, caso exista	58
A.4	Tela inicial da aplicação do evento EBL 2019	59
A.5	Tela que permite selecionar trabalhos que devem aparecer juntos	60
A.6	Exibição das preferências dos trabalhos que devem aparecer juntos . . .	60

LISTA DE TABELAS

2.1	Trabalhos relacionados ao PPSTC	12
3.1	Trabalhos e temas	13
5.1	Resultados computacionais da F1 para as instâncias baseadas no XV LARS	35
5.2	Resultados computacionais da F2 para as instâncias baseadas no XV LARS	36
5.3	Resultados computacionais do HILS para as instâncias baseadas no XV LARS	37
5.4	Resultados computacionais da F1 para as instâncias baseadas na XIX EBL	38
5.5	Resultados computacionais da F2 para as instâncias baseadas na XIX EBL	38
5.6	Resultados computacionais do HILS para as instâncias baseadas na XIX EBL	39
5.7	Resultados computacionais da F1 para as instâncias baseadas no LI SBPO	39
5.8	Resultados computacionais da F2 para as instâncias baseadas no LI SBPO	40
5.9	Resultados computacionais do HILS para as instâncias baseadas no LI SBPO	40
5.10	Resultado computacionais para as instâncias baseadas no XV LARS . .	42
5.11	Resultados computacionais para as instâncias baseadas na XIX EBL . .	45
5.12	Resultado computacionais para as instâncias baseadas no LI SBPO . .	47
5.13	Resultado computacionais para as instâncias reais	49

GLOSSÁRIO

AGA	Algoritmo genético de agrupamento
BCP	<i>Branch, cut and price</i>
BPP	<i>Bin packing problem</i>
DC	<i>Designer</i> combinatorial
EBL	Conferência Brasileira de Lógica
ILS	<i>Iterated local search</i>
LARS	<i>Latin American Robotics Symposium</i>
LB	<i>Lower bound</i>
MPI	Modelo de Programação Inteira
OC	Otimização combinatória
PC	<i>Programme chair</i>
PPGC	Problema de Particionamento de Grafos em <i>Cliques</i>
PPSTC	Problema de programação de sessões técnicas de conferências
PA	Preferência dos Apresentadores
PP	Preferência dos Participantes
RVND	<i>Randomized variable neighborhood descent</i>
SA	<i>Simulated annealing</i>
SBPO	Simpósio Brasileiro de Pesquisa Operacional
SOM	<i>Self Organizing Map</i>
SP	<i>Set Partitioning</i>
UB	<i>Upper bound</i>
VND	<i>Variable Neighborhood Descent</i>

Capítulo 1

Introdução

1.1 Definição do tema

Anualmente, milhões de pessoas frequentam eventos acadêmicos (científicos) ao redor do mundo. Usualmente, esses eventos são chamados de conferências (terminologia adotada neste trabalho), *workshops*, simpósios, reuniões, entre outros, e podem durar de um dia a uma semana. Um típico formato de conferência é composto por sessões técnicas (em que os participantes apresentam brevemente seus trabalhos), plenárias (ex. apresentadores *keynote*), cursos de curta duração (ex. tutoriais), mesas-redondas, apresentação de posters e interações sociais (ex. *coffee-breaks*, jantar de gala).

Uma sessão técnica é constituída por um conjunto de trabalhos, geralmente com tópicos em comum. Organizar tais sessões é um problema de otimização combinatória (OC). Assim, o problema de programação de sessões técnicas em conferências (PPSTC) consiste em alocar trabalhos a sessões levando em consideração a relação entre os tópicos dos trabalhos, a preferência dos participantes e/ou dos palestrantes.

Na literatura, é possível encontrar diversas abordagens para esse problema. Em sua maioria, tais abordagens são apresentadas na forma de algoritmos exatos (garante a melhor solução) ou heurísticas (garante soluções sub-ótimas). No geral, heurísticas são utilizadas quando a abordagem exata para o problema é inviável por questões de tempo de execução ou recursos computacionais.

Uma abordagem não anula a outra. Pode-se utilizá-las em conjunto de forma híbrida, em que um modelo exato pode incluir uma heurística, assim como, uma heurística pode incluir um modelo exato.

Este trabalho apresenta duas abordagens exatas e uma híbrida para o PPSTC. As próximas seções descrevem a justificativa, objetivos e estrutura do trabalho.

1.2 Justificativa

Por se tratar de um problema combinatório, gerar a programação de sessões técnicas é uma tarefa árdua de ser realizada manualmente, e mais difícil ainda se adicionarmos restrições que são intrínsecas ao problema como, por exemplo, evitar que trabalhos de um mesmo autor sejam alocados em sessões que ocorrem em um mesmo período de tempo.

Problemas envolvendo programação de eventos são estudados desde a década de 1980. Na literatura, é possível encontrar diferentes abordagens, no entanto, cada uma delas têm sua própria forma de mitigar o problema como distingue bem Vangerven et al (2018). Para Thompson (2002), de um modo geral, pode-se classificar tais abordagens em duas categorias. A primeira consiste em uma abordagem baseada na perspectiva dos apresentadores (PA), na qual podem ser levadas em consideração suas preferências e disponibilidades. A segunda baseia-se na perspectiva dos participantes (PP) do evento, na qual leva-se em consideração, principalmente, as suas preferências por certas palestras disponibilizadas no evento.

Nas diversas abordagens analisadas, notou-se que cada uma delas possui aspectos bastante distintos no que diz respeito à forma de tratar o problema. Além disso, algumas dependem de informações que devem ser fornecidas pelos participantes e/ou apresentadores. Isso pode levar a resultados insatisfatórios. É bastante otimista supor que todos os participantes irão submeter suas preferências, assim, se apenas uma minoria o fizer, isso pode acarretar em uma programação insatisfatória para a maioria dos participantes. Vangerven et al (2018) discutem também outras limitações e desvantagens desse tipo de abordagem, que depende das preferências dos participantes, e cita o fato de que sessões com trabalhos com tópicos muito divergentes podem ser formadas.

Além disso, também cita que a programação pode ser mais afetada pelos participantes que optarem por mais apresentações, o que pode impactar negativamente aqueles que selecionaram um pequeno número de apresentações.

Devido às particularidades de cada conferência, é virtualmente impossível criar uma modelagem que abarque os requisitos de todas. No entanto, uma quantidade considerável desses eventos possuem características semelhantes, isso motivou a desenvolver uma versão básica, mas realista para o problema e que tenha ampla aplicabilidade. Espera-se encorajar organizadores de conferências a empregar uma ferramenta baseada em otimização e também que outros pesquisadores possam comparar suas metodologias e sugerir melhorias e generalizações. Uma iniciativa de desenvolver uma formulação genérica para o problema de programação de sessões técnicas já foi realizada por Edis e Sancar Edis (2013), mas seu modelo leva em consideração apenas um tópico por artigo, e cada sessão só pode ter artigos do mesmo tópico. Essa e outras suposições limitam um pouco a aplicabilidade de sua abordagem.

Neste trabalho, tem-se o interesse em formular o problema de forma simultânea, levando em consideração tanto a perspectiva dos apresentadores quanto a dos participantes. A ideia é focar nos tópicos dos trabalhos, buscando trabalhos fortemente relacionados e alocá-los em uma mesma sessão, de forma semelhante ao que Tanaka et al (2002) realizaram com palavras-chave.

Da perspectiva dos apresentadores (autores), é proibido que trabalhos de um mesmo autor sejam alocados em sessões paralelas (que ocorrem ao mesmo tempo). Além disso, embora não seja explicitamente considerada nesta pesquisa, a formulação proposta do problema é geral e flexível o suficiente para lidar com situações em que os apresentadores desejam especificar sua disponibilidade, bem como com aspectos adicionais relacionados às preferências dos participantes.

Duas formulações matemáticas são propostas para resolver o problema em questão, que é \mathcal{NP} -hard. Porém, dada a sua complexidade, o emprego de tais formulações se limita a instâncias com um número limitado de artigos, tipicamente bem inferior ao número de trabalhos apresentados em eventos reais. Desta forma, para resolver instâncias de porte maior, propõe-se um algoritmo híbrido que combina elementos das meta-heurísticas *Iterated Local Search* (ILS) e *Simulated Annealing* (SA), além de

fazer uso de modelos de programação inteira baseado em particionamento de conjuntos e alocação generalizada.

1.3 Objetivos

1.3.1 Objetivo geral

Propor formulações matemáticas e um algoritmo híbrido para o Problema de Programação de Sessões Técnicas de Conferências.

1.3.2 Objetivos específicos

- Formular o problema.
- Propor modelos exatos para o problema.
- Gerar instâncias artificiais para o problema.
- Avaliar a eficiência dos modelos exatos em relação ao tamanho das instâncias.
- Desenvolver um algoritmo híbrido capaz de lidar com instâncias de grande porte.
- Avaliar a qualidade das soluções do algoritmo híbrido em relação aos modelos exatos.
- Comparar a solução do algoritmo híbrido com eventos passados organizados de forma manual.
- Aplicar algoritmo híbrido em novos eventos reais.
- Avaliar os resultados obtidos.

1.4 Estrutura do trabalho

O restante deste trabalho está organizado da seguinte maneira:

- O Capítulo 2 apresenta uma revisão dos trabalhos relacionados ao tema de programação de sessões técnicas em conferências.
- O Capítulo 3 descreve o problema e algumas definições.

- O Capítulo 4 explica em detalhes a abordagem adotada e o algoritmo proposto, discutindo seus componentes.
- O Capítulo 5 apresenta os experimentos computacionais conduzidos os resultados
- O Capítulo 6 apresenta a conclusão da pesquisa.

Capítulo 2

Trabalhos relacionados

Nesta seção, é feito um breve resumo dos principais trabalhos relacionados ao problema de programação técnica de conferências. Existem diversas variantes do problema, mas, em suma, todas compartilham um único objetivo, maximizar a satisfação de quem participa do evento, garantindo que seus frequentadores tenham a oportunidade de acompanhar suas áreas e/ou apresentações de interesse.

Vale destacar que algumas variantes do problema alocam diretamente apresentações em faixas de horário desconsiderando sessões como sendo um conjunto de trabalhos. Outras, porém, alocam as sessões que podem ser construídas pela solução proposta ou não. Ou seja, alguns trabalhos da literatura consideram que as sessões já foram construídas em uma etapa anterior e se preocupam apenas em alocar essas sessões. Outras, no entanto, incorporam a tarefa de atribuir trabalhos a sessões como parte da solução.

Eglese e Rand (1987) são autores de um dos trabalhos pioneiros em tentar resolver o problema de alocação de sessões técnicas. A abordagem deles é voltada as preferências dos participantes e tem como objetivo fazer com que eles assistam ao máximo possível de apresentações que lhes interessem. Para isso, a abordagem proposta toma como entrada um conjunto de formulários fornecidos pelos participantes contendo 4 opções de apresentações e mais um de reserva em ordem de prioridade. Nessa versão do problema, o intuito é fazer com que as apresentações sejam atribuídas a períodos de

tempo e a salas de modo que o custo das violações das preferências sejam minimizados. Para cada opção de um participante, é atribuído um peso diferente de acordo com a prioridade de sua escolha. Note que as apresentações são alocadas sem levar em conta seus tópicos. Para gerar a solução do problema, algumas restrições foram levadas em consideração, como o número de salas disponíveis e se a sala continha os recursos de que o palestrante necessitava. Nesta versão do problema, é permitido que as apresentações sejam repetidas ao longo da programação gerada. O trabalho apresenta uma formulação exata, mas que, devido ao tamanho do problema, não seria praticável seu uso tendo em vista as limitações computacionais da época e, por isso, decidiu-se resolvê-lo através de uma heurística. A heurística possui duas partes, a primeira encontra uma solução e a segunda faz uma melhoria através de um algoritmo de *annealing*. A solução proposta foi aplicada em um evento real contendo 15 seminários.

Sampson e Weiss (1995) tomaram como base a mesma versão do problema apresentado por Eglese e Rand (1987) e o expandiram para que fosse levada em consideração a capacidade das salas. O problema é dividido em duas partes principais. A primeira é atribuir de forma viável as sessões ofertadas às salas de acordo com suas capacidades. A segunda é atribuir os participantes às sessões. Os autores fazem um paralelo com o problema de matrícula em uma instituição de ensino, em que um conjunto de estudantes devem optar por certas disciplinas ofertadas respeitando o limite máximo de alunos. Nesse caso, as disciplinas seriam as sessões, estudantes os participantes e os professores seriam os apresentadores. Uma heurística é proposta para resolver ambas as partes do problema. As duas partes do problema são resolvidas de forma simultânea. A heurística atribui sessões a períodos de tempo e salas enquanto atribui participantes a sessões. O procedimento apresentado foi testado utilizando instâncias geradas de forma aleatória.

Em Tanaka et al (2002), a versão do problema considera que sessões devem conter trabalhos que abordam assuntos semelhantes. O nível de relação entre trabalhos é medido a partir das palavras-chave que eles possuem em comum. Essa abordagem é voltada às preferências dos apresentadores, uma vez que evita a simultaneidade de sessões, ou seja, que autores apresentem mais de um trabalho ao mesmo tempo em sessões distintas. Os autores já realizaram uma tentativa de resolver esse problema

em Mori e Tanaka (2002) utilizando um algoritmo genético de agrupamento (AGA). No entanto, o alto custo computacional e a dificuldade em tratar sessões de tamanhos diferentes os fizeram propor essa nova solução. O problema é resolvido através de um *Self Organizing Map* (SOM). O SOM foi criado nos anos 80 e é um paradigma de rede neural de aprendizado não supervisionado. O método proposto utilizou os dados de uma conferência que contava com 313 trabalhos e um total de 86 palavras-chaves sumarizadas em categorias de forma manual.

Thompson (2002) utiliza uma abordagem orientada as preferências dos participantes. Um de seus objetivos nesse trabalho é mostrar que esse tipo de abordagem pode aumentar a satisfação dos participantes se comparada com a abordagem voltada às preferências dos apresentadores. A versão do problema abordado é semelhante a utilizada por Eglese e Rand (1987). Assim como Sampson e Weiss (1995) ele considera que as salas têm capacidade, mas, neste caso, elas podem ter tamanhos diferentes. Outra diferença é considerar que as salas não estão disponíveis o tempo todo. Os participantes podem fornecer de 0 a 8 preferências. As preferências não possuem prioridade ou pesos. Uma heurística construtiva, juntamente com *Simulated Annealing*, é utilizada para resolver o problema. São realizados três experimentos construídos a partir de uma conferência real. Em um desses experimentos, é solicitado para que alguns alunos tentem gerar a programação. A programação também é gerada de forma aleatória e utilizando a heurística. A análise realizada mostra que tanto a solução dada pelos alunos quanto de forma aleatória têm praticamente a mesma efetividade. Já a heurística apresentou um resultado superior as outras.

Potthoff e Munger (2003) têm por objetivo distribuir sessões ao longo da programação visando balancear os temas ao longo do evento. Cada sessão contém 3 ou 4 trabalhos pertencentes a uma mesma área. Todos os trabalhos são agrupados em uma etapa anterior, assim, o método de solução apresentado busca alocar as sessões às faixas de horários. É considerado que os apresentadores podem desempenhar outros papéis no evento como, por exemplo, *chair* de uma outra sessão, portanto, deve-se fazer com que ele não esteja em mais de um lugar ao mesmo tempo, seja como apresentador, co-autor do trabalho ou algum outro compromisso que possa vir a ter no evento. O problema é resolvido através de um modelo de programação inteira (MPI). Para testar

a efetividade do modelo, foi utilizado uma edição já ocorrida (2001) do *Annual Meetings of the Public Choice Society*. Em um trabalho posterior, Potthoff e Brams (2007), o método foi aplicado para gerar a programação das edições de 2005 e 2006 do evento citado. Na programação dessas duas edições uma nova restrição foi levada em conta, a disponibilidade dos autores.

Sampson (2004) traz uma discussão aprofundada sobre programação de conferência baseada em preferências. O artigo estabelece a teoria por trás dessa abordagem. Além disso, é feito um comparativo do problema de programação de sessões de conferências com problemas similares encontrados na literatura como *exam scheduling* e *course timetabling*. É apresentada uma formulação baseada em Eglese e Rand (1987), porém, com algumas diferenças como a não repetição de sessões, a capacidade das salas e das sessões são consideradas compostas por vários trabalhos. Para resolver o problema, foi utilizado uma versão modificada do *Simulated Annealing*.

Nicholls (2007) tem o objetivo de alocar os trabalhos às sessões com base nas preferências dos apresentadores levando em consideração algumas preferências implícitas dos participantes. As preferências implícitas citadas são atendidas evitando alocar sessões com um mesmo tema de forma paralela, dando a oportunidade dos participantes terem mais chance de acompanhar trabalhos da sua área de interesse. Apesar do artigo tratar das preferências dos apresentadores, apenas alguns deles solicitaram algum dia ou horário específico. Uma heurística é apresentada para a solução do problema. A heurística não tem uma função objetiva explícita, apenas algumas restrições simples. Ela tenta acomodar o máximo possível de preferências dos apresentadores e alocar todas as sessões. Ela se preocupa mais em encontrar uma solução viável. Existe um programa assistente, que ajuda o *programme chair* (PC) a organizar a sessão. A heurística foi aplicada em um evento real. As sessões foram construídas em uma etapa anterior pelos *track chairs*. O PC pode alterar as sessões e checar se houve conflito de autores. Ele recebe dos apresentadores suas preferências por dias e/ou horários. Essas preferências são acomodadas utilizando a heurística.

Ibrahim et al (2008) utilizam um *designer* combinatorial (DC) para construir uma programação cujos temas sejam balanceados ao longo do evento. *Designer* combinatorial é um método de selecionar subconjuntos de um conjunto finito de tal modo que

algumas restrições ou condições sejam satisfeitas. Nessa versão do problema, considera-se que cada sessão tem 3 áreas. As sessões devem ser alocadas em períodos de tempo ao longo de k dias.

No trabalho de Edis e Sancar Edis (2013) é proposto um modelo de programação inteira genérico. A proposta é atribuir os tópicos às sessões e preenchê-las com os trabalhos pertencentes a cada tópico. Os tópicos que são atribuídos a cada sessão são dados de entrada. Considera-se que cada trabalho tem apenas um tópico. O modelo é resolvido utilizando duas funções objetivas. A primeira busca minimizar o número de sessões paralelas com o mesmo tópico. A segunda busca minimizar a diferença de tamanho entre sessões paralelas. Essa abordagem pode ser considerada como sendo orientada às preferências dos apresentadores uma vez que não recebe nenhum dado de entrada vindo dos participantes além de evitar que autores apresentem trabalhos em sessões paralelas. Uma instância hipotética é utilizada para avaliar a efetividade da solução.

Quesnelle e Steffy (2015) buscam encontrar um escalonamento que seja factível em relação ao apresentador e à logística das salas, mas tentando minimizar o conflito das preferências dos participantes do evento. É proposto um modelo de programação inteira. O modelo garante que os autores de um mesmo trabalho não tenham conflitos de horário com sessões paralelas. Essa abordagem também leva em consideração as salas. No caso, busca-se fazer com que as sessões sejam alocadas em salas adequadas. As salas possuem um período de disponibilidade. Nessa versão do problema, não existe o conceito de tópicos, apenas de apresentações e portanto sessões podem conter temas mistos. Quando o modelo é executado sem as preferências dos participantes seu tempo de execução é bastante rápido. No entanto, quando as preferências são consideradas o desempenho cai consideravelmente mesmo utilizando uma máquina com *hardware* de alto desempenho. Uma justificativa para isso que os autores fornecem é a simetria do problema. Por exemplo, duas salas podem ter as mesmas características, gerando assim, uma simetria. Uma dualização de uma restrição é realizada gerando uma melhora na performance em 75%.

Vangerven et al (2018) têm por objetivo maximizar a satisfação dos participantes e, em segundo plano, minimizar saltos entre sessões. Saltos entre sessões referem-se ao

ato de um participante ter que trocar de sala durante uma sessão para que possa acompanhar as apresentações de seu interesse. Segundo os autores, isso gera incômodo tanto para os participantes quanto para os apresentadores já que as apresentações podem ser interrompidas pelo constante fluxo de pessoas entrando e saindo da sala. Essa abordagem pode ter como consequência a criação de sessões incoerentes no que diz respeito aos temas dos trabalhos, uma vez que as preferências dos participantes é que definem os trabalhos que irão compor a sessão. No entanto, como geralmente as preferências tendem a ter trabalhos de temas similares, as sessões resultantes podem apresentar uma certa coerência. Outra consequência é que as preferências dos participantes não são consideradas igualmente. Participantes com muitas opções podem ter um maior impacto na programação do evento do que outros que escolhem menos. Na abordagem apresentada, a disponibilidade dos autores também é levada em consideração. O método de solução proposto é uma abordagem hierárquica de três fases. A primeira fase utiliza um modelo de programação inteira com o objetivo de maximizar as preferências dos participantes. A segunda fase possui duas versões, uma utilizando um modelo de programação inteira e a outra utilizando uma heurística. Nessa fase, o objetivo é minimizar os saltos entre sessões. A terceira fase busca minimizar as violações de disponibilidade dos apresentadores. O método proposto foi aplicado em 4 eventos reais.

A Tabela 2.1 apresenta um breve resumo dos métodos e abordagens utilizados pelos trabalhos apresentados. Nota-se que a maioria dos trabalhos se valem de modelos de programação inteira (MPI) ou Heurística. Ademais, a abordagem orientada as preferências dos apresentadores (PA), apesar de ser maioria, não difere tanto da quantidade de trabalhos que utilizam a abordagem orientada as preferências dos participantes (PP). No total, 7 trabalhos utilizaram a abordagem PA e 5 trabalhos a PP.

Tabela 2.1: Trabalhos relacionados ao PPSTC

Trabalho	Ano	Método	Abordagem
Eglese e Rand	1987	Heurística	PP
Sampson e Weiss	1995	Heurística	PP
Tanaka et al	2002	SOM	PA
Mori e Tanaka	2002	AGA	PA
Thompson	2002	Heurística + SA	PP
Potthoff e Munger	2003	MPI	PA
Sampson	2004	SA	PP
Nicholls	2007	Heurística	PA
Ibrahim et al	2008	DC	PA
Edis e Sancar Edis	2013	MPI	PA
Quesnelle e Steffy	2015	MPI	PA
Vangerven et al	2018	MPI e Heurística	PP

Capítulo 3

Definição do problema e formulação matemática

Dada uma conferência, o objetivo do PPSTC é encontrar uma solução para a programação das sessões técnicas. Uma sessão técnica geralmente é constituída por dois ou mais trabalhos que possuem temas semelhantes. Devido ao grande número de trabalhos presentes nas conferências, é comum que várias sessões ocorram ao mesmo tempo, ou seja, de forma paralela. Dependendo do evento, os trabalhos podem ter um ou mais temas associados. Na abordagem adotada neste trabalho, são justamente esses temas que servem de parâmetro para medir a semelhança de dois trabalhos e, conseqüentemente, o benefício de estarem juntos em uma mesma sessão.

Tabela 3.1: Trabalhos e temas

Trabalho	Tema
Trabalho 1	A, B, C
Trabalho 2	A, B, D
Trabalho 3	D, E, F
Trabalho 4	D, E, F

Na Tabela 3.1, é apresentado um conjunto de trabalhos e seus respectivos temas. Note que os trabalhos 1 e 2 possuem dois temas em comum, o trabalho 2, por sua vez, possui um tema em comum com os trabalhos 3 e 4. Já os trabalhos 3 e 4 possuem os três

temas em comum. Embora o trabalho 2 tenha um tema em comum com os trabalhos 2 e 3, o benefício dele estar em uma mesma sessão com o trabalho 1 é maior do que estar com os demais, já que ambos possuem mais temas em comum. Considerando que esses quatro trabalhos devam ser alocados em duas sessões, a alocação ideal seria colocar os trabalhos 1 e 2 em uma sessão e o 3 e 4 em outra.

A programação das sessões técnicas deve levar em consideração algumas restrições. Uma delas é a restrição de que um apresentador, que possua mais de um trabalho, não tenha seus trabalhos alocados em sessões paralelas, visto que não poderá atender aos dois compromissos ao mesmo tempo. Se a sessão contendo os trabalhos 1 e 2, e a sessão contendo os trabalhos 3 e 4 fossem paralelas e os trabalhos 2 e 3 tivessem o mesmo autor, haveria então uma solução inviável visto que violaria a restrição de um apresentador poder apenas ter um trabalho por sessão paralela. O ideal, então, seria que o trabalho 2 ou 3 fosse alocado em alguma outra sessão com horário diferente. Outras restrições para o problema serão apresentadas na Seção 3.1.

Para os quatro trabalhos, a tarefa é trivial, no entanto, suponha um evento com cerca de 200 trabalhos, o número de combinações possíveis e inviabilidades que podem ocorrer por conta de algumas restrições tornam o problema muito difícil de ser resolvido de forma manual.

No restante do capítulo, será apresentado a definição formal do problema e duas versões de formulações matemáticas para o problema.

3.1 Definição do Problema

Dado um conjunto de trabalhos $P = \{1, \dots, n\}$, um conjunto de dias D , um conjunto de horários H não sobrepostos (ex., 08:00–10:00 e 10:00–12:00 são dois horários distintos e sem sobreposição), um conjunto de sessões S , um conjunto de temas T e um conjunto de autores A . Assume-se que o apresentador de um dado trabalho é um de seus autores.

Além disso, sejam $P_a \subseteq P$ o conjunto de trabalhos do autor $a \in A$, e $P_t \subseteq P$ o conjunto de trabalhos que possuem o tema $t \in T$. Uma sessão $s \in S$ acontece no dia $d_s \in D$, no horário $h_s \in H$, e possui limites mínimo e máximo de trabalhos, dados por

m_s e M_s , respectivamente. Para $d \in D$ e $h \in H$, defina: $S_d = \{s \in S : d_s = d\}$ como o conjunto de sessões do dia d ; $S_h = \{s \in S : h_s = h\}$ o conjunto de sessões do horário h ; e $S_d^h = \{s \in S : d_s = d \wedge h_s = h\}$ o conjunto de sessões (paralelas) do dia d e horário h .

O problema de programação de sessões técnicas de conferências proposto consiste em alocar trabalhos, elementos do conjunto P , a sessões, elementos do conjunto S , satisfazendo as seguintes restrições.

- Um trabalho $i \in P$ deve ser atribuído a exatamente uma sessão.
- O número mínimo (m_s) e máximo (M_s) de trabalhos alocados em uma sessão não pode ser violado, desde que a sessão seja utilizada. Note que sessões vazias são permitidas.
- Se ao menos um trabalho $p \in P$ de um dado autor tiver sido alocado em uma sessão $s \in S$, não pode haver outro trabalho do mesmo autor alocado em uma sessão paralela.
- Para evitar que vários trabalhos de um mesmo tema $t \in T$ sejam alocados em sessões paralelas, considera-se um limite superior M_H de trabalhos com o tema t que podem ser alocados em um mesmo horário $h \in H$ e em um mesmo dia $d \in D$, independente da sessão.
- Da mesma forma, com o intuito de evitar uma concentração de trabalhos com o mesmo tema $t \in T$ em um mesmo dia, define-se um limite superior M_D de trabalhos com o mesmo tema que podem ser alocados em um mesmo dia $d \in D$.

O objetivo é encontrar uma programação que maximize o benefício total, em que o benefício de alocar dois trabalhos juntos $i, j \in P$ na mesma sessão é dado pelo valor $b_{ij} \in \mathbb{R}$. Assume-se que o conjunto de sessões possui capacidade suficiente para acomodar todos os trabalhos.

O PPSTC é um problema \mathcal{NP} -difícil (Preposição 1), pois o Problema de Particionamento de Grafos em *Cliques* (PPGC) estudado por Grötschel e Wakabayashi (1990) pode ser a ele reduzido.

Proposição 1. *O problema PPSTC é \mathcal{NP} -difícil.*

Prova. Seja $G = (V, E)$ uma instância PPSTC. Na redução, cria-se uma instância I do PPSTC, na qual existem $|V|$ sessões, cada uma denotando uma partição (possivelmente vazia), e $|V|$ trabalhos, cada um representando um vértice distinto de G . As sessões não são simultâneas, isto é, elas não ocorrem em paralelo, e não possuem restrições de limite (i.e., $m_s = 1$ e $M_s = |V|$). Cada trabalho tem um autor e um tema distinto, assim, qualquer alocação satisfaz as restrições relativas aos autores e temas. Por fim, o benefício de alocar dois trabalhos i e j em uma mesma sessão é igual ao peso da aresta de G cujos extremos são os vértices representados por i e j . Sendo assim, a solução ótima de I contém uma alocação de trabalhos a sessões que é equivalente a uma solução ótima do PPSTC para G . \square

Uma vez encontrada uma solução viável, assume-se que os trabalhos podem ser alocados às suas respectivas sessões. Além disso, os organizadores também podem optar por atribuir um tema à sessão (ex., levando em consideração a moda dos temas presentes na sessão).

Embora não tenham sido considerados explicitamente as restrições de disponibilidade dos apresentadores, destaca-se que isso pode ser facilmente tratado na prática. Se os organizadores desejarem considerar essa restrição, basta remover as variáveis da formulação que violam a disponibilidade do autor, evitando que seus trabalhos sejam alocados em horários indesejados. Além disso, as preferências dos participantes podem ser tratadas de diferentes maneiras, dependendo da política adotada pelos organizadores. Por exemplo, em vez de considerar os tópicos, eles podem levar em consideração o número de participantes interessados em assistir as apresentações de um par de trabalhos como a medida de relação entre eles. Alternativamente, pode-se introduzir um conjunto de *soft constraints* (restrições que podem ser violadas) na forma de um objetivo secundário com o intuito de minimizar a violação do número de apresentações selecionadas por um participante que foram designadas para sessões paralelas, ou mesmo adicionando *hard constraints* (restrições invioláveis) para evitar que isso não ocorra de forma alguma.

3.2 Formulações Matemáticas

A presente seção descreve as duas formulações matemáticas propostas, aqui denominadas de F1 e F2.

3.2.1 Formulação F1

Para a formulação F1, foram utilizados quatro tipos de variáveis de decisão. Defina x_{ij}^s como sendo uma variável binária que assumirá o valor 1 quando dois trabalhos $i, j \in P$, $i < j$, estiverem juntos em uma mesma sessão $s \in S$ e y_i^s como sendo uma variável binária que assumirá o valor 1 apenas quando o trabalho $i \in P$ estiver na sessão $s \in S$. Por último, seja v_a^s uma variável binária com valor 1 quando o autor $a \in A$ estiver na sessão $s \in S$ e u^s uma variável binária que assumirá o valor 1 quando a sessão $s \in S$ for utilizada. O modelo de programação inteira linear pode ser escrito da seguinte forma:

$$\text{Max} \sum_{i \in P} \sum_{\substack{j \in P \\ i < j}} \sum_{s \in S} b_{ij} x_{ij}^s \quad (3.1)$$

Sujeito a:

$$\sum_{s \in S} y_i^s = 1, \quad \forall i \in P \quad (3.2)$$

$$m_s u^s \leq \sum_{i \in P} y_i^s \leq M_s u^s, \quad \forall s \in S \quad (3.3)$$

$$|P_a| v_a^s \geq \sum_{i \in P_a} y_i^s, \quad \forall a \in A, \forall s \in S \quad (3.4)$$

$$\sum_{s \in S_d^h} v_a^s \leq 1, \quad \forall a \in A, \forall h \in H, \forall d \in D \quad (3.5)$$

$$\sum_{i \in P_t} \sum_{s \in S_d^h} y_i^s \leq M_H, \quad \forall t \in T, \forall h \in H, \forall d \in D \quad (3.6)$$

$$\sum_{i \in P_t} \sum_{s \in S_d} y_i^s \leq M_D, \quad \forall t \in T, \forall d \in D \quad (3.7)$$

$$x_{ij}^s \leq y_i^s, \quad \forall i \in P, \forall j \in P, i < j, \forall s \in S \quad (3.8)$$

$$x_{ij}^s \leq y_j^s, \quad \forall i \in P, \forall j \in P, i < j, \forall s \in S \quad (3.9)$$

$$x_{ij}^s + 1 \geq y_i^s + y_j^s, \quad \forall i \in P, \forall j \in P, i < j, \forall s \in S \quad (3.10)$$

$$x_{ij}^s \in \{0, 1\}, \quad \forall i \in P, \forall j \in P, i < j, \forall s \in S \quad (3.11)$$

$$y_i^s \in \{0, 1\}, \quad \forall i \in P, \forall s \in S \quad (3.12)$$

$$u^s \in \{0, 1\}, \quad \forall s \in S. \quad (3.13)$$

$$v_a^s \in \{0, 1\}, \quad \forall a \in A, \forall s \in S \quad (3.14)$$

A função objetivo (3.1) maximiza o benefício de dois trabalhos estarem juntos em uma mesma sessão. As restrições (3.2) garantem que todos os trabalhos sejam alocados a exatamente uma sessão. As restrições (3.3) impõem os limites mínimo e máximo da sessão, se, e somente se, a sessão for usada. As restrições (3.4) e (3.5) evitam que dois ou mais trabalhos de um mesmo autor sejam alocado em sessões paralelas. As restrições (3.6) e (3.7) limitam a quantidade de artigos de um mesmo tema em sessões paralelas e em um mesmo dia, respectivamente. As restrições (3.8)–(3.10) fazem a relação entre as variáveis de decisão. As restrições (3.11)–(3.14) determinam a natureza das variáveis de decisão.

É possível calcular o número mínimo (L_{\min}) e máximo (L_{\max}) de arestas em uma solução em função do número de artigos e das capacidades das sessões disponíveis, de uma maneira similar à empregada por (Bulhões et al, 2017). Uma aresta representa dois trabalhos alocados em uma mesma sessão. Com isso, pode-se incluir a seguinte restrição com o intuito de fortalecer a relaxação linear de F1:

$$L_{\min} \leq \sum_{i \in P} \sum_{\substack{j \in P \\ i < j}} \sum_{s \in S} x_{ij}^s \leq L_{\max} \quad (3.15)$$

3.2.2 Formulação F2

A ideia da formulação F2 é primeiramente resolver o PPSTC como um problema de *clustering*. Nesse contexto, considera-se que diferentes tipos de *clusters* de artigos podem ser construídos. Cada tipo de *cluster* $k \in K$ é caracterizado por um número

mínimo (m_K) e máximo (M_k) de trabalhos que podem ser parte de um tipo de *cluster*. Portanto, o conjunto K deriva dos limites m_s e M_s que são impostos por cada tipo de sessão $s \in S$ cujos limites são associados a cada tipo de *cluster* $k \in K$, i.e., $m_s = m_k$ e $M_s = M_k$.

Seja z_{ij} uma variável binária que assumirá valor 1 quando os trabalhos $i, j \in P, i < j$ estiverem juntos no mesmo *cluster* e w_{ij}^k uma variável binária que assumira valor 1 quando os trabalhos $i, j \in P, i \leq j$, estiverem juntos no mesmo *cluster* do tipo $k \in K$ que seja liderado pelo trabalho i . O líder de um *cluster* é o trabalho de menor índice nele contido. A formulação F2 pode ser escrita da seguinte forma:

$$\text{Max} \sum_{i \in P} \sum_{\substack{j \in P \\ i < j}} b_{ij} z_{ij} \quad (3.16)$$

Sujeito a:

$$z_{ij} + z_{jv} - z_{iv} \leq 1, \quad \forall i \in P, \forall j \in P, \forall v \in P, i < j < v, \quad (3.17)$$

$$z_{ij} - z_{jv} + z_{iv} \leq 1, \quad \forall i \in P, \forall j \in P, \forall v \in P, i < j < v \quad (3.18)$$

$$-z_{ij} + z_{jv} + z_{iv} + \sum_{k \in K} w_{vv}^k \leq 1, \quad \forall i \in P, \forall j \in P, \forall v \in P, i < j < v \quad (3.19)$$

$$\sum_{k \in K} \sum_{\substack{i \in P \\ i \leq j}} w_{ij}^k = 1, \quad \forall j \in P \quad (3.20)$$

$$\sum_{j \in P} w_{jj}^k \leq |S_k|, \quad \forall k \in K \quad (3.21)$$

$$m_K w_{ii}^k \leq \sum_{\substack{j \in P \\ i \leq j}} w_{ij}^k \leq k w_{ii}^k, \quad \forall k \in K, \forall i \in P \quad (3.22)$$

$$w_{ij}^k \leq w_{ii}^k, \quad \forall k \in K, \forall i \in P, \forall j \in P, i < j \quad (3.23)$$

$$L_i \leq \sum_{i \in P} \sum_{\substack{j \in P \\ i < j}} z_{ij} \leq L_s \quad (3.24)$$

$$\sum_{k \in K} w_{jj}^k + z_{ij} \leq 1, \quad \forall i \in P, \forall j \in P, i < j \quad (3.25)$$

$$\sum_{k \in K} w_{jj}^k + \sum_{\substack{i \in P \\ i < j}} z_{ij} \geq 1, \quad \forall j \in P \quad (3.26)$$

$$z_{ij} \geq \sum_{k \in K} w_{ij}^k, \quad \forall i \in P, \forall j \in P, i < j \quad (3.27)$$

$$z_{ij} \in \{0, 1\}, \quad \forall i \in P, \forall j \in P, i < j \quad (3.28)$$

$$w_{ij}^k \in \{0, 1\}, \quad \forall i \in P, \forall j \in P, i < j, \forall k \in K. \quad (3.29)$$

A função objetivo (3.16) busca maximizar o benefício total da alocação dos trabalhos. As restrições (3.17)–(3.19) proíbem a existência de um caminho induzido formado por três vértices. Em outras palavras, elas garantem que, dado que os trabalhos i e j e os trabalhos j e v estão na mesma sessão, então os trabalhos i e v também devem estar na mesma sessão. A inserção do termo $\sum_{k \in K} w_{vv}^k$ nas restrições (3.19) é um fortalecimento proposto em Bulhões et al (2017). As restrições (3.20) obrigam que todos os trabalhos sejam alocados em exatamente um *cluster*. As restrições (3.21) determinam que existem, no máximo, $|S_k|$ *clusters* do tipo k , enquanto que as restrições (3.22) garantem que os tamanhos máximo e mínimo de um determinado tipo de *cluster* sejam respeitados. As restrições (3.23) asseguram que o trabalho i só pode representar o trabalho j caso i seja líder de *cluster*. As restrições (3.24) limitam a quantidade de arestas da solução. As restrições (3.25)–(3.26) determinam quando um trabalho é líder de *cluster*. As restrições (3.27) forçam a aresta $\{i, j\}$ a existir caso o trabalho i seja o líder do *cluster* que contém o trabalho j . As restrições (3.28)–(3.29) estão associadas ao domínio das variáveis de decisão.

Uma solução dada pela formulação (3.16)–(3.29) induz a um *clustering* de trabalhos. No entanto, essa formulação é incompleta porque as três últimas restrições definidas na Seção 3.1, i.e., as relacionadas a autores e temas podem ser violadas. Daqui em diante elas serão referidas como restrições adicionais.

As seguintes restrições são adicionadas para tornar F2 completa:

$$\sum_{\{i,j\} \in E} z_{ij} \leq |E| - 1, \quad \forall E \in \mathcal{E} \quad (3.30)$$

em que $\mathcal{E} = \{E_1, E_2, \dots, E_q\}$ é uma coleção que contém todos os q *clusterings* que satisfazem (3.16)–(3.29) mas violam ao menos uma das restrições adicionais.

Mais precisamente, cada elemento $E \in \mathcal{E}$ é um conjunto de arestas que determina

um certo *clustering* inviável. Devido ao número de restrições (3.30) ser potencialmente exponencial, na prática elas são adicionados sob demanda como cortes *lazy* conforme será descrito no capítulo a seguir.

Capítulo 4

Algoritmos propostos

Nesta seção, serão abordados dois algoritmos propostos para o PPSTC. O primeiro é um algoritmo exato sobre a formulação F2 e o segundo trata-se de um algoritmo híbrido.

4.1 Branch-and-cut com separação *lazy*

Foi implementado um algoritmo *branch-and-cut* (BC) sobre F2 adicionando as restrições (3.30) de forma *lazy*. Em particular, cada vez que uma solução incumbente é encontrada, ou seja, uma solução melhor que a atual, é verificado se o *clustering* induzido por tal solução pode gerar uma solução viável com relação as restrições adicionais. Isto é realizado resolvendo um problema de separação, conforme descrito a seguir.

4.1.1 Problema de separação

Seja $\mathcal{C} = \{C_1, C_2, \dots, C_l\}$ uma coleção de *clusters* de uma solução incumbente encontrada por F2. Defina $\mathcal{C}_a \subseteq \mathcal{C}$ como o conjunto de *clusters* que contém os trabalhos do autor $a \in A$ e $S_j \subseteq S$ como o conjunto de sessões em que o *cluster* $C_j \in \mathcal{C}$ pode ser alocado. A quantidade de trabalhos do *cluster* $C_j \in \mathcal{C}$ que possuem o tema $t \in T$ é denotado por Q_j^t . Por fim, a variável binária ξ_j^s indica se o *cluster* $C_j \in \mathcal{C}$ foi alocado

na sessão $s \in S$. Portanto, o modelo associado ao problema de separação pode ser escrito da seguinte forma:

$$\text{Min } 0 \tag{4.1}$$

Sujeito a:

$$\sum_{s \in S_j} \xi_j^s = 1, \quad \forall C_j \in \mathcal{C} \tag{4.2}$$

$$\sum_{C_j \in \mathcal{C}: s \in S_j} \xi_j^s \leq 1, \quad \forall s \in S \tag{4.3}$$

$$\sum_{C_j \in \mathcal{C}_a} \sum_{s \in S_d^h \cap S_j} \xi_j^s \leq 1, \quad \forall a \in A, \forall d \in D, \forall h \in H \tag{4.4}$$

$$\sum_{C_j \in \mathcal{C}} \sum_{s \in S_d^h \cap S_j} Q_j^t \xi_j^s \leq M_H, \quad \forall t \in T, \forall d \in D, \forall h \in H \tag{4.5}$$

$$\sum_{C_j \in \mathcal{C}} \sum_{s \in S_d \cap S_j} Q_j^t \xi_j^s \leq M_D, \quad \forall t \in T, \forall d \in D \tag{4.6}$$

$$\xi_j^s \in \{0, 1\}, \quad \forall C_j \in \mathcal{C}, \forall s \in S_j. \tag{4.7}$$

O modelo acima apenas verifica se o *clustering* encontrado satisfaz as restrições adicionais, deste modo uma função objetivo não se faz necessária. As restrições (4.2) impõem que cada *cluster* deve ser alocado a apenas uma sessão. As restrições (4.3) garantem que no máximo um *cluster* possa ser alocado a cada sessão. As restrições (4.4) evitam que um mesmo autor tenha trabalhos alocados em sessões paralelas. As restrições (4.5) e (4.6) limitam a quantidade de trabalhos de um mesmo tema por horário e dia, respectivamente. As restrições (4.7) especificam o domínio das variáveis.

A seguir, será demonstrado que o problema de separação é \mathcal{NP} -completo por meio de uma redução do *bin packing problem* (BPP).

Proposição 2. *O problema de separação definido por (4.1)–(4.7) é \mathcal{NP} -completo.*

Prova. Seja I uma instância do BPP com n_i itens i_1, i_2, \dots, i_{n_i} , pesos positivos w_1, w_2, \dots, w_{n_i} e n_b bins de capacidade B . Defina uma instância I' associada com

o problema de separação tal que: $D = \{d_1\}$, $H = \{h_1, h_2, \dots, h_{n_b}\}$, $T = \{t_1\}$, $\mathcal{C} = \{C_1, C_2, \dots, C_{n_i}\}$, $M_H = B$ e $M_D = Bn_b$. O *cluster* C_i satisfaz $Q_i^{t_1} = w_i$; isto é, tem w_i trabalhos, todos sobre o tema t_1 . O conjunto de autores A é definido de tal modo que cada trabalho tem apenas um autor e cada autor só possui um trabalho, o que torna as restrições (4.4) redundantes. Cada horário $h \in H$ tem n_i sessões e cada sessão $s \in S$ satisfaz $m_s = 1$ e $M_s = B$. Portanto, um *cluster* C_i pode ser alocado em qualquer sessão. A ideia da redução é que cada horário está associado a um *bin* e cada *cluster* está associado a um item. É fácil ver que I é uma *YES-instance* (instância de um problema de decisão cuja resposta é sim) se, e somente se I' é uma *YES-instance*. \square

Note que o problema de separação pode ser facilmente generalizado para modelar aspectos adicionais que podem ser considerados pelos organizadores tal como as preferências dos participantes. Por exemplo, dado um subconjunto de apresentações que um participante deseja assistir, pode-se incluir uma função objetivo que visa minimizar a ocorrência dessas apresentações em sessões paralelas. Na prática, isto seria o mesmo que considerar uma função objetivo secundária para o problema completo já discutido na Seção 3.1. Idealmente, seria preferível que as apresentações selecionadas pelo participante não fossem alocadas em sessões paralelas, para que o participante pudesse assistir a todas. Se os organizadores desejarem impor essa situação, um conjunto de *hard constraints* similar a (4.4) poderiam ser adicionadas com o intuito de lidar com essa política ao invés de utilizar uma função objetivo.

4.1.2 Limites primais

Visando melhorar a performance do *branch-and-cut*, o algoritmo híbrido proposto (Seção 4.2) foi utilizado para gerar o limite primal inicial.

4.2 O algoritmo híbrido HILS

A seguir, será apresentado o algoritmo híbrido proposto para a solução do problema de programação de sessões técnicas de eventos acadêmicos, denominado HILS. O mé-

todo baseia-se no *framework* da meta-heurística *iterated local search* (ILS) (Lourenço et al, 2010) e utiliza em sua busca local o procedimento baseado em *Variable Neighborhood Descedent* (VND) (Mladenović e Hansen, 1997), além de um procedimento inspirado na meta-heurística *simulated annealing* (SA) (Kirkpatrick et al, 1983) como critério de aceitação de uma solução a ser perturbada.

Algoritmo 1 HILS

```

1: Procedimento HILS( $I_{Max}$ ,  $I_{ILS}$ ,  $\tau_0$ ,  $\alpha$ )
2:  $f^* \leftarrow 0$ 
3:  $s^* \leftarrow 0$ 
4:  $ClusterPool \leftarrow \emptyset$ 
5: para  $i := 1$  até  $I_{Max}$  faça
6:    $s \leftarrow \text{GeraSolucaoInicial}()$ 
7:    $s' \leftarrow s$ 
8:    $s'' \leftarrow s$ 
9:    $iter_{ILS} \leftarrow 0$ 
10:   $\tau \leftarrow \tau_0$ 
11:  enquanto  $iter_{ILS} \leq I_{ILS}$  faça
12:     $s \leftarrow \text{BuscaLocal}(s)$ 
13:     $ClusterPool \leftarrow \text{AtualizaClusterPool}(s, ClusterPool)$ 
14:     $SolutionStore \leftarrow \text{AtualizaSolutionStore}(s)$ 
15:     $\Delta \leftarrow f(s) - f(s')$ 
16:    se  $\Delta > 0$  então
17:       $s' \leftarrow s$ 
18:       $s'' \leftarrow s$ 
19:       $iter_{ILS} \leftarrow 0$ 
20:    senão
21:       $x \in [0, 1]$ 
22:      se  $\tau > 0$  e  $x < e^{-\Delta/\tau}$  então
23:         $s' \leftarrow s$ 
24:        senão
25:           $s' \leftarrow s''$ 
26:        fim se
27:      fim se
28:       $s \leftarrow \text{Perturba}(s')$ 
29:       $iter_{ILS} \leftarrow iter_{ILS} + 1$ 
30:       $\tau \leftarrow \tau \times \alpha$ 
31:    fim enquanto
32:    se  $f(s') > f^*$  e  $\text{VerificaViabilidade}(s') = \text{true}$  então
33:       $s^* \leftarrow s'$ 
34:       $f^* \leftarrow f(s')$ 
35:    fim se
36:  fim para
37:  $s^* \leftarrow \text{SP}(s^*, ClusterPool)$ 
38: retorne  $s^*$ 
39: fim HILS.

```

Conforme apresentado no Algoritmo 1, o algoritmo híbrido *mult-start* executa I_{Max} iterações (linhas 5–36). Para cada iteração uma solução inicial é gerada por um método guloso e aleatório (linha 6). Em seguida, o laço interno do algoritmo (linhas 11–31)

tenta melhorar a solução inicial gerada aplicando um procedimento de busca local (linha 12) combinado com um mecanismo de perturbação (linha 28). Esse laço continua até atingir I_{ILS} iterações sem melhora. Note que sempre que uma solução de melhor custo é encontrada o contador $iter_{ILS}$ é reiniciado (linhas 16–19).

Na etapa de perturbação, também são consideradas soluções que não levam a melhoria. Assim, para criar uma maior diversificação, ela pode ser feita sobre uma solução de pior custo, de maneira análoga a meta-heurística SA. As soluções que não levam a melhoria serão aceitas com probabilidade em função do parâmetro τ (linhas 22–26). Inicialmente, τ é inicializado como sendo τ_0 (linha 10) e posteriormente reduzido a cada iteração em uma razão $\Delta \in (0, 1)$ (linha 30), diminuindo conseqüentemente a probabilidade de aceitação das soluções de pior benefício. Caso a solução de piora não seja aceita, a solução corrente é restaurada para a melhor solução da iteração (linha 25).

No HILS, cada solução é composta por um conjunto de *clusters*, e cada *cluster* contém um conjunto de artigos. Os *clusters* são uma representação de cada sessão do problema. Assim, para cada sessão existe um *cluster* equivalente com a mesma capacidade. A cada vez que uma busca local é realizada, os *clusters* associados à solução podem ser adicionados a um *pool* de *clusters* (linha 13). O método adiciona apenas *clusters* que não foram gerados por outras soluções, ou seja, *clusters* diferentes dos que já estão no *pool*. A solução retornada pela busca local é salva em uma estrutura de dados denominada *SolutionStore* (linha 13). Ela armazena o valor da solução bem como a configuração dos *clusters* que a gerou. Assim, se durante a fase de busca local essa configuração for encontrada novamente, a solução é imediatamente retornada pois já é conhecido que essa é a melhor solução da região sendo explorada.

Ao final de cada iteração do laço principal, a solução é atualizada em caso de melhoria e se for viável (linhas 32–34). Neste caso, utiliza-se o mesmo procedimento descrito na Seção 4.1.1 tendo como entrada os *clusters* encontrados pelo algoritmo naquela iteração.

Quando ambos os laços são finalizados, o algoritmo realiza sua última etapa. Trata-se de um procedimento baseado em particionamento de conjuntos (Seção 4.2.1) para encontrar a melhor combinação dos *clusters* que foram armazenados no *pool* (linha

37). Novamente, o modelo de separação é chamado dinamicamente para checar se as soluções incumbentes encontradas são viáveis.

4.2.1 Abordagem baseada em particionamento de conjuntos

Conforme discutido anteriormente, ao final do algoritmo HILS foi utilizada uma abordagem baseada em particionamento de conjuntos (linha 37). Considere C um conjunto de *clusters*, cada qual com um benefício $b'_j, j \in C$, associado e K um conjunto dos tamanhos de cada *cluster* (equivalente ao tamanho máximo das sessões). Para cada trabalho $i \in P$, existe um subconjunto C_i de *clusters* que contém aquele trabalho, e para cada tamanho de *cluster* existe um conjunto S_k de sessões de tamanho k e um conjunto C_k de *clusters* aptos a serem alocados em uma sessão de tamanho $k \in K$.

O intuito do uso do particionamento de conjuntos é encontrar os *clusters* armazenados no *pool* de forma a maximizar o benefício total. Assuma λ_j com uma variável binária que indica quando um *cluster* $j \in C$ é utilizado.

$$\text{Max} \sum_{j \in C} b'_j \lambda_j \quad (4.8)$$

Sujeito a:

$$\sum_{j \in C_i} \lambda_j = 1, \quad \forall i \in P \quad (4.9)$$

$$\sum_{j \in C_k} \lambda_j \leq S_k, \quad \forall k \in K \quad (4.10)$$

$$\sum_{j \in C} \lambda_j \leq |S| \quad (4.11)$$

$$\lambda_j \in \{0, 1\}, \quad \forall j \in C. \quad (4.12)$$

A função objetivo (4.8) é maximizada pela escolha dos *clusters* de melhor benefício. As restrições (4.9) obrigam que cada trabalho só deve estar em um *cluster*. As restrições (4.10) limitam a quantidade de *clusters* de tamanho $k \in K$ que podem entrar na solução. A restrição (4.11) limita a quantidade de *clusters*. As restrições (4.12) determinam a natureza da variável de decisão.

4.2.2 Geração da solução inicial

A solução inicial é gerada a partir de um algoritmo guloso e aleatório. Inicialmente, para cada *cluster* é atribuído um trabalho de forma aleatória. Os outros trabalhos são inseridos iterativamente em uma segunda etapa. Nessa etapa, cada trabalho é alocado no *cluster* que obtiver o melhor benefício, ou seja, que tenha a melhor soma *intracluster*. Se ao final existirem *clusters* que não atendem o limite mínimo, os trabalhos contidos neles são removidos e realocados novamente. Se mesmo assim ainda existirem *clusters* com trabalhos abaixo do limite, esses trabalhos serão realocados em um *cluster dummy* com peso negativo. O peso negativo faz com que durante a fase de busca local os trabalhos contidos no *cluster dummy* sejam alocados em outros *clusters*.

4.2.3 Busca local

A fase de busca local é realizada através de um método baseado no procedimento VND com escolha aleatória na ordem das vizinhanças. Cada vizinhança realiza um tipo de movimento que leva ao vizinho viável de maior custo. Quando uma vizinhança não melhora a solução incumbente, uma nova é selecionada aleatoriamente.

Algoritmo 2 BuscaLocal

```

1: Procedimento BuscaLocal( $s$ )
2: IniciarEstruturaAuxiliar()
3: IniciarListaDeVizinhanças(LV)
4: enquanto LV  $\neq$  0 faça
5:   Escolha uma vizinhança  $V^{(n)} \in$  LV aleatoriamente
6:   Encontre o melhor vizinho  $s'$  de  $s \in V^{(n)}$ 
7:   se  $f(s') > f(s)$  então
8:      $s \leftarrow s'$ 
9:   se PossuiSolucao( $s$ , SolutionStore) então
10:    retorne ObterSolucao( $s$ , SolutionStore)
11:   fim se
12:   senão
13:     Remova  $V^{(n)}$  de LV
14:   fim se
15:   AtualizarEstruturaAuxiliar()
16: fim enquanto
17: retorne  $s$ 
18: fim BuscaLocal

```

O pseudocódigo do Algoritmo 2 apresenta o procedimento de busca local utilizado. O primeiro passo é inicializar as estruturas auxiliares (linha 2) e a lista contendo as

vizinhanças (linha 3). O laço principal (linhas 4–16) inicia-se pela escolha aleatória de uma vizinhança (linha 5) seguida pela busca do melhor vizinho (linha 6). Enquanto existirem vizinhanças a serem exploradas em LV o laço é repetido. Se for encontrado uma melhora para a solução (linhas 7–14), a solução corrente é atualizada para a encontrada (linha 8), caso contrário a vizinhança é removida da lista LV (linha 13). Quando uma melhora na solução é encontrada, é realizada uma busca na *SolutionStore* (linha 9) para saber se a solução (mais precisamente a configuração atual dos *clusters*) é conhecida. Se essa solução for conhecida, significa que as vizinhanças já foram exploradas nessa região, e portanto o seu resultado já é conhecido e deste modo já possível retorná-lo (linha 10) evitando iterações desnecessárias. Vale lembrar que a *SolutionStore* é atualizada no Algoritmo 1 (linha 14). Antes de terminar o laço principal as estruturas auxiliares são atualizadas (linha 15). Por fim, ao término do laço principal, caso nenhuma solução de melhora esteja contida na *SolutionStore*, a melhor solução encontrado pela busca local é retornada.

Utilizou-se a estratégia de melhor aprimorante (*best improvement*) durante a busca em cada vizinhança. Além disso, para melhorar a eficiência da busca local, foi empregada a abordagem conhecida como *don't look bits* (Bentley, 1992), que examina apenas movimentos que possuem potencial de melhora, descartando outros em que já se sabe de antemão que não levarão a melhora por já terem sido explorados anteriormente.

4.2.4 Estruturas de Vizinhança

Para a resolução do problema utilizou-se duas estruturas de vizinhança que serão descritas a seguir:

- *Move* — $N^{(1)}$: Um *cluster* que contenha ao menos um trabalho é selecionado e o trabalho é realocado em outro *cluster* que ainda esteja abaixo de sua capacidade máxima.
- *Swap* — $N^{(2)}$: Dois *clusters* não vazios trocam um trabalho entre si.

4.2.5 Estruturas de Dados Auxiliares (EDAs)

Para avaliar os movimentos em tempo constante, foram empregadas estruturas de dados auxiliares que armazenam algumas informações como o benefício de um *cluster* ao se remover/inserir um trabalho. As atualizações de tais estruturas são feitas em tempo $O(n^2)$. Como o tamanho de cada vizinhança é da ordem de $O(n^2)$ e cada movimento é avaliado em tempo $O(1)$, então a complexidade total do procedimento de busca local é $O(n^2)$. As EDAs utilizadas são listadas a seguir:

ClusterPool — armazena todos os *clusters* encontrados durante a execução do algoritmo para que ao final sejam passados para o SP.

SolutionStore — esta estrutura armazena uma lista de soluções da busca local. A solução é representada por um valor inteiro, que armazena o resultado da solução, e por um *hash* formado a partir dos *clusters* que compõem a solução. Assim quando uma solução é salva no *SolutionStore*, é gerado um *hash* a partir da configuração atual dos *clusters* que é salvo juntamente com o valor da solução. Quando deseja-se saber se uma solução está armazenada no *SolutionStore*, gera-se um *hash* da solução e a compara com as que estão armazenadas no *SolutionStore*.

Benefícios [][] — *array* bidimensional que armazena o benefício de um trabalho estar junto com outro em uma mesma sessão. Os benefícios são calculados apenas uma vez antes de iniciar o HILS.

MovimentosInserção [][] — armazena o benefício de inserir um trabalho em um *cluster*.

MovimentosTroca [][] — armazena os custos em realizar uma troca entre dois trabalhos em *clusters* diferentes.

MovimentoInfo — Estrutura de dados que armazena a informação relacionada entre um *Cluster* e um trabalho. Para cada trabalho existe uma instância dessa estrutura. Ela armazena as seguintes informações:

- referência ao trabalho;
- referência ao *cluster* cujo trabalho pertence;
- benefício de remover o trabalho do *cluster*;

- índice do *cluster* que traz maior benefício ao mover o trabalho referenciado até ele (*Move*);
- valor do benefício ao mover o trabalho referenciado até o *cluster* indicado no item anterior;
- índice do *cluster* e do trabalho que trás maior benefício ao realizar um movimento de troca (*Swap*);
- valor do benefício da melhor troca.

Através da estrutura *MovimentoInfo* é possível identificar qual o melhor vizinho de cada vizinhança e quais os *clusters* envolvidos em cada ação, o que possibilita atualizar apenas as EDAs referentes aos *clusters* que são impactados por cada uma das operações.

4.2.6 Mecanismos de perturbação

Dois mecanismos de perturbação são utilizados. A cada iteração é selecionada aleatoriamente uma das seguintes perturbações:

- *Multiple moves* — $P^{(1)}$: De modo similar ao procedimento da busca local, no entanto, tanto o *cluster* de origem quanto o de destino são escolhidos de forma aleatória e o procedimento se repete de 2 a 5 vezes.
- *Multiple swaps* — $P^{(2)}$: De modo similar ao procedimento da busca local, no entanto, ambos os *clusters* são escolhidos de forma aleatória e o procedimento se repete de 2 a 5 vezes.

Capítulo 5

Resultados

Os experimentos computacionais foram realizados em um computador com processador Intel[®] Xeon[®] CPU E5-2650 com 2.20 GHz, 128.0 GB de memória RAM com sistema operacional *Ubuntu* 16.04.6. A linguagem de programação utilizada nas implementações dos modelos matemáticos e do algoritmo híbrido foi C++ (g++ 5.4.0). O *solver* de programação linear inteira adotado foi o CPLEX versão 12.7.

5.1 Calibração do algoritmo híbrido

Com o intuito de melhorar o desempenho do algoritmo híbrido, realizou-se uma série de testes variando os seguintes parâmetros: I_{Max} , I_{ILS} , τ_0 , α .

Nesse processo o SP executado ao final do algoritmo foi desativado para que fosse possível identificar os parâmetros que geram os melhores resultados.

Os parâmetros foram calibrados um de cada vez. Para cada um deles, foi testado um conjunto de valores, do menor para o maior. Considerou-se que para um parâmetro ser melhor que outro, o valor da solução deve ser no mínimo 0.5% melhor que a solução do parâmetro anterior. Esse limite ajuda a evitar que o tempo de execução aumente sem ganho expressivo na solução.

O primeiro parâmetro foi o I_{ILS} . Para a calibração desse parâmetro, foi desativado também o SA e foi atribuído o valor 1 para I_{Max} . O valor de I_{ILS} é dado por

$\max(100, n|P|)$, onde n é o multiplicador que foi alterado na calibração. Os valores testados para n são 1, 1.5 e 2. O valor que apresentou o melhor custo desempenho em termos de valor obtido e tempo de execução foi $n = 1$. Portanto, para a calibração dos demais parâmetros, fixou-se $n = 1$.

Para o parâmetro I_{Max} , os valores adotados foram 20, 30 e 40. Com o parâmetro $I_{Max} = 40$ observou-se que os valores obtidos eram maiores na maioria das instâncias.

Para a calibração do SA, o valor de τ_0 foi fixado em 1000 e foi feita uma variação no parâmetro α com os seguintes parâmetros: 0.4, 0.6 e 0.8. O valor de $\alpha = 0.4$ foi o que apresentou os melhores resultados.

Portanto, os parâmetros adotados nos resultados a seguir são: $I_{ILS} = \max(100, |P|)$, $I_{Max} = 40$, $\tau_0 = 1000$ e $\alpha = 0.4$.

O valor do benefício b_{ij} de se alocar dois trabalhos $i \in P$ e $j \in P$ na mesma sessão é determinado pela quantidade de temas em comum entre eles, a saber: 1, 10 e 100 para um, dois e três temas em comum, respectivamente.

As instâncias utilizadas no processo de calibração são as mesmas apresentadas na seção a seguir.

5.2 Resultados obtidos em instâncias de pequeno porte

Para testar os modelos e o algoritmo híbrido, criou-se um gerador de instâncias que toma como base os dados de algum evento. O gerador recebe como parâmetro um valor que representa a porcentagem de proporcionalidade que se deseja criar a instância. Portanto, é possível criar instâncias menores mantendo aspectos da instância real. Os temas de cada trabalho são definidos com base na probabilidade de um trabalho ter um certo número de temas de acordo com a instância real bem como a probabilidade de um tema aparecer em algum trabalho. O mesmo método de geração de temas foi utilizado para gerar os autores de cada trabalho. Embora as formulações e o algoritmo híbrido sejam capazes de realizar a programação levando em consideração todos os autores de um trabalho, leva-se em conta apenas os apresentadores, já que é o que ocorre geralmente na prática. Para as instâncias geradas, o apresentador é selecionado

de forma aleatória entre os autores do trabalho. Como critério de parada para os modelos, foi estabelecido o limite de três horas para o tempo de execução. Para o SP do HILS foi estabelecido o limite de 5 minutos.

Os testes foram realizados tomando como base os dados dos eventos *XV Latin American Robotics Symposium* (LARS) 2018, XIX Conferência Brasileira de Lógica (EBL) 2019 e o LI Simpósio Brasileiro de Pesquisa Operacional (SBPO) 2019. Para averiguar o desempenho das formulações e do algoritmo híbrido, foram gerados 18 conjuntos de instâncias de tamanhos diferentes para o EBL e LARS. Já para o SBPO foram gerados 10 conjuntos. Para cada conjunto, foram geradas 5 instâncias. Portanto, ao todo foram geradas 230 instâncias. Como o EBL e o LARS apresentam menos de 100 trabalhos, geramos instância que vão de 30% até 200% do tamanho do evento real. O SBPO, por sua vez, é um evento que contém quase 300 trabalhos e por isso geramos instâncias que vão de 10% até 100% do tamanho real. O tamanho de cada instância representa a quantidade de trabalhos existentes nela. O valor percentual representa o tamanho das instâncias em relação ao LARS (86 trabalhos), EBL (93 trabalhos), SBPO (288 trabalhos).

As Tabelas 5.1, 5.2, 5.4, 5.5, 5.7, 5.8 apresentam os resultados obtidos com as formulações F1 e F2. A primeira coluna indica o tamanho das instâncias em termos da quantidade de trabalhos. A coluna UB_R indica o *upper bound* da raiz. As colunas UB e LB indicam, respectivamente, o *upper bound* e *lower bound* da solução. A coluna Gap(%) exibe os *gaps* entre o LB e o UB. T(s) indica o tempo de execução em segundos. As linhas com '-' indicam as instâncias em que nenhuma solução viável foi encontrada dentro do tempo limite. As Tabelas 5.3, 5.6, 5.9 apresentam os resultados obtidos pelo algoritmo HILS, onde $Gap(\%)_{UB}$ indica o desvio percentual entre o valor do benefício encontrado pelo algoritmo híbrido (LB) e o melhor UB obtido pelas formulações. Para cada instância, o HILS foi executado 10 vezes. Os valores apresentados para o HILS são uma média dessas execuções.

Nos resultados da Tabela 5.1, a formulação F1 obteve todos os ótimos para os dois primeiros conjuntos de instâncias. Já no terceiro conjunto, não consegue convergir dentro do tempo limite. No quarto conjunto, em uma instância, não foi possível encontrar nenhuma solução viável e nas demais não houve convergência. No quinto conjunto, foi

Tabela 5.1: Resultados computacionais da F1 para as instâncias baseadas no XV LARS

Tamanho	UB_R	UB	LB	Gap(%)	T(s)
26 (30%)	674,90	672,00	672,00	0,00	86,28
	467,00	463,00	463,00	0,00	33,67
	672,80	616,00	616,00	0,00	65,02
	386,25	373,00	373,00	0,00	21,14
	486,12	481,00	481,00	0,00	33,89
34 (40%)	265,00	264,00	264,00	0,00	3452,55
	361,00	360,00	360,00	0,00	2225,71
	393,30	389,00	389,00	0,00	81,12
	202,00	201,00	201,00	0,00	1516,95
	299	298,00	298,00	0,00	8836,21
43 (50%)	320,75	315,00	320,50	1,75	10800,00
	337	336,00	337,00	0,30	10800,00
	433	430,00	433,00	0,70	10800,00
	310	308,00	310,00	0,65	10800,00
	280	279,00	280,00	0,36	10800,00
52 (60%)	686	669,00	686,00	2,54	10800,00
	365,33	360,00	365,00	2,54	10800,00
	-	-	-	-	10800,00
	396,87	391,00	396,87	1,50	10800,00
	329	325,00	326,00	1,23	10800,00
60 (70%)	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00
	422	417,00	422,00	1,20	10800,00

Tabela 5.2: Resultados computacionais da F2 para as instâncias baseadas no XV LARS

Tamanho	UB_R	UB	LB	Gap(%)	T(s)
26 (30%)	677,77	672,00	672,00	0,00	84,07
	466,52	463,00	463,00	0,00	105,96
	624,16	616,00	616,00	0,00	36,65
	377,79	373,00	373,00	0,00	66,74
	483,27	481,00	481,00	0,00	99,33
34 (40%)	265,82	264,00	264,00	0,00	9279,05
	361,14	360,00	360,00	0,00	1442,65
	390,53	339,00	339,00	0,00	1599,19
	201,80	201,00	201,00	0,00	235,39
	299,14	298,00	298,00	0,00	227,11
43 (50%)	318,96	315,00	318,94	1,25	10800,00
	337,56	336,00	336,00	0,00	515,51
	-	-	-	-	10800,00
	309,92	308,00	308,00	0,00	5476,96
	-	-	-	-	10800,00
52 (60%)	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00
	393,87	391,00	391,00	0,00	3097,78
	-	-	-	-	10800,00
60 (70%)	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00

Tabela 5.3: Resultados computacionais do HILS para as instâncias baseadas no XV LARS

Tamanho	LB	Gap_{UB}(%)	T(s)
26 (30%)	672,00	0,00	1,65
	463,00	0,00	2,10
	616,00	0,00	1,85
	373,00	0,00	1,95
	481,00	0,00	1,81
34 (40%)	264,00	0,00	3,60
	360,00	0,00	3,10
	389,00	0,00	3,09
	201,00	0,00	3,01
	298,00	0,00	3,08
43 (50%)	315,00	0,00	4,67
	336,00	0,00	4,58
	430,00	0,00	5,22
	308,00	0,00	4,56
	279,00	0,00	5,57
52 (60%)	670,00	0,00	7,40
	361,00	0,00	7,99
	576,00	-	7,79
	391,00	0,00	7,23
	325,00	0,00	6,94
60 (70%)	447,00	-	9,66
	464,00	-	11,55
	395,00	-	10,75
	400,00	-	8,70
	417,00	0,00	9,42

Tabela 5.4: Resultados computacionais da F1 para as instâncias baseadas na XIX EBL

Tamanho	UB_R	UB	LB	Gap(%)	T(s)
28 (30%)	314,00	312,00	314,00	0,00	80,39
	353,37	330,00	330,00	0,00	353,37
	367,00	364,00	364,00	0,00	66,88
	654,75	553,00	553,00	0,00	22,62
	475	467,00	467,00	0,00	18,80
37 (40%)	339,00	339,00	336,00	0,00	3465,99
	375,00	373,00	375,00	0,25	10800,00
	517,42	499,00	517,42	0,00	10800,00
	373,00	363,00	363,00	0,00	356,03
	123	123,00	123,00	0,00	404,94
47 (50%)	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	743,52	689,00	743,52,00	7,91	10800,00
	-	-	-	-	-

Tabela 5.5: Resultados computacionais da F2 para as instâncias baseadas na XIX EBL

Tamanho	UB_R	UB	LB	Gap(%)	T(s)
28 (30%)	315,26	312,00	312,00	0,00	84,07
	337,58	330,00	330,00	0,00	105,96
	365,00	364,00	364,00	0,00	36,65
	558,86	553,00	553,00	0,00	66,74
	468,5	467,00	467,00	0,00	99,33
37 (40%)	-	-	-	-	10800,00
	397,16	373,00	397,16	6,48	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00
	123,00	123,00	123,00	0,00	404,94
47 (50%)	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00

Tabela 5.6: Resultados computacionais do HILS para as instâncias baseadas na XIX EBL

Tamanho	LB	Gap_{UB}(%)	T(s)
28 (30%)	312,00	0,00	2,28
	330,00	0,00	2,08
	364,00	0,00	2,00
	553,00	0,00	2,46
	467,00	0,00	3,05
37 (40%)	339,00	0,00	3,66
	373,00	0,00	3,34
	499,00	0,00	30,65
	363,00	0,00	3,30
	123,00	0,00	4,91
47 (50%)	463,00	-	5,36
	608,00	-	6,62
	781,00	-	4,81
	689,00	-	5,08
	671,00	0,00	5,29

Tabela 5.7: Resultados computacionais da F1 para as instâncias baseadas no LI SBPO

Tamanho	UB_R	UB	LB	Gap(%)	T(s)
29 (10%)	280,00	262,00	262,00	0,00	86,51
	253,00	181,00	253,00	39,78	10800,00
	244,00	208,00	244,00	17,31	10800,00
	280,00	226,00	280,00	23,89	10800,00
	280,00	217,00	280,00	29,03	10800,00
56 (20%)	560,00	524,00	560,00	6,87	10800,00
	560,00	506,00	560,00	10,67	10800,00
	560,00	506,00	560,00	10,67	10800,00
	560,00	515,00	560,00	8,74	10800,00
	560	497,00	560,00	12,68	10800,00
86 (30%)	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-

Tabela 5.8: Resultados computacionais da F2 para as instâncias baseadas no LI SBPO

Tamanho	UB_R	UB	LB	Gap(%)	T(s)
29 (10%)	272,75	262,00	262,00	0,00	5,21
	195,05	181,00	195,00	7,73	10800,00
	209,54	208,00	208,00	0,00	209,54
	234,74	226,00	226,00	0,00	234,74
	219,00	217,00	217,00	0,00	392,03
56 (20%)	546,00	524,00	546,00	4,26	10800,00
	532,00	506,00	532,00	5,14	10800,00
	542,07	506,00	542,00	7,12	10800,00
	551,00	515,00	551,00	6,99	10800,00
	538,00	497,00	538,00	8,25	10800,00
86 (30%)	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00
	-	-	-	-	10800,00

Tabela 5.9: Resultados computacionais do HILS para as instâncias baseadas no LI SBPO

Tamanho	LB	Gap_{UB}(%)	T(s)
29 (10%)	262,00	0,00	2,29
	181,00	0,00	2,09
	208,00	0,00	1,78
	226,00	0,00	1,80
	217,00	0,00	2,05
56 (20%)	524,00	0,00	16,10
	506,00	0,00	6,36
	506,00	0,00	222,36
	515,00	0,00	306,75
	497,00	0,00	15,25
86 (30%)	652,00	0,00	11,24
	688,00	0,00	12,12
	706,00	0,00	12,53
	697,00	0,00	11,52
	670,00	0,00	11,19

encontrada uma solução em apenas uma instância. Para os resultados da Tabela 5.4, a formulação F1 conseguiu resolver todas as instâncias para o primeiro conjunto. No segundo conjunto, só conseguiu resolver três instâncias de forma exata. No terceiro conjunto, apenas em uma instância a formulação foi capaz de encontrar uma solução. Para os resultados da Tabela 5.7, apenas em uma instância a formulação F1 conseguiu convergir dentro do tempo limite dentro do primeiro conjunto. Por conseguinte, no segundo conjunto, a formulação atingiu o tempo limite em todas as instâncias. Ademais, no terceiro conjunto, não foi possível encontrar nenhuma solução viável.

Para a formulação F2, nos resultados da Tabela 5.2, os dois primeiros conjuntos foram resolvidos de forma ótima. No terceiro conjunto, em apenas duas instâncias o ótimo foi encontrado. Em uma, o tempo limite foi atingido e, nas outras duas, não foi possível encontrar uma solução viável dentro do tempo limite. A formulação no quarto conjunto conseguiu resolver apenas uma instância. Já no quinto conjunto, nenhuma instância foi resolvida. Nos resultados da Tabela 5.5, a formulação F2 alcançou todos os ótimos para o primeiro conjunto. Apenas duas instâncias foram resolvidas no segundo conjunto, sendo uma delas de forma ótima. No entanto, no terceiro conjunto, nenhuma solução viável foi encontrada. Os resultados da Tabela 5.8 mostram que a formulação só resolveu 4 instâncias com otimalidade no primeiro conjunto. Para o segundo conjunto, nenhuma solução convergiu dentro do tempo limite e, no terceiro conjunto, nenhuma solução viável foi encontrada. Note que apesar da formulação F1 ter encontrado mais soluções viáveis, ela não domina totalmente F2, já que, em alguns casos, F2 foi capaz de encontrar soluções ótimas enquanto F1 atingiu o tempo limite.

Para os resultados das Tabelas 5.3, 5.6 e 5.9, o HILS conseguiu atingir todos os ótimos conhecidos além de resolver as instâncias que F1 e F2 não foram capazes.

5.2.1 Resultados obtidos em instâncias de grande porte

Para este segundo teste, utilizaram-se todas as instâncias geradas. Novamente, o HILS foi executado 10 vezes em cada instância. Os resultados obtidos foram comparados com aqueles obtidos pelo algoritmo *branch-cut-and-price* (BCP) proposto por Bulhões et al (2019). As Tabelas 5.10, 5.11 e 5.12 exibem o resultado comparativo entre o BCP e a média dos resultados obtidos pelo HILS. Como é possível notar, nas

instâncias em que o ótimo é atingido, o HILS conseguiu alcançar a maioria deles, mais precisamente, dentre as 147 instâncias em que o ótimo é conhecido o HILS conseguiu obter a otimalidade em 132 delas. A Tabela 5.13 apresenta o resultado das instâncias reais de cada evento.

Tabela 5.10: Resultado computacionais para as instâncias baseadas no XV LARS

Tamanho	BCP			HILS		
	UB	Gap(%)	T(s)	Sol.	Gap _{UB} (%)	T(s)
26 (30%)	672,00	0,00	0,72	672,00	0,00	1,65
	463,00	0,00	0,38	463,00	0,00	2,10
	616,00	0,00	0,20	616,00	0,00	1,85
	373,00	0,00	0,23	373,00	0,00	1,95
	481,00	0,00	0,15	481,00	0,00	1,81
34 (40%)	264,00	0,00	17,23	264,00	0,00	3,60
	360,00	0,00	2,01	360,00	0,00	3,10
	389,00	0,00	0,39	389,00	0,00	3,09
	201,00	0,00	0,36	201,00	0,00	3,01
	298,00	0,00	0,31	298,00	0,00	3,08
43 (50%)	315,00	0,00	1,34	315,00	0,00	4,67
	336,00	0,00	3,37	336,00	0,00	4,58
	430,00	0,00	0,95	430,00	0,00	5,22
	308,00	0,00	1,11	308,00	0,00	4,56
	279,00	0,00	0,87	279,00	0,00	5,57
52 (60%)	670,00	0,00	1,97	670,00	0,00	7,40
	361,00	0,00	1,08	361,00	0,00	7,99
	576,00	0,00	2,89	576,00	0,00	7,79
	391,00	0,00	4,36	391,00	0,00	7,23
	325,00	0,00	2,72	325,00	0,00	6,94
60 (70%)	447,00	0,00	43,43	447,00	0,00	9,66
	464,00	0,00	22,38	464,00	0,00	11,55
	395,00	0,00	2,61	395,00	0,00	10,75
	400,00	0,00	3,63	400,00	0,00	8,70
	417,00	0,00	28,19	417,00	0,00	9,42
69 (80%)	474,00	0,00	231,33	474,00	0,00	14,73
	459,00	0,00	3,01	459,00	0,00	14,86
	442,00	0,00	47,43	442,00	0,00	11,05

Continua na próxima página

Tamanho	BCP		HILS			
	UB	Gap(%)	T(s)	Sol.	Gap _{UB} (%)	T(s)
	432,00	0,00	112,24	432,00	0,00	14,07
	440,00	0,00	63,61	440,00	0,00	13,00
	483,00	0,00	88,87	483,00	0,00	17,61
	394,00	0,00	24,42	392,00	0,51	15,84
73 (90%)	450,00	0,00	9,72	450,00	0,00	16,66
	450,00	0,00	85,70	450,00	0,00	15,86
	421,00	0,00	43,68	421,00	0,00	17,01
	455,00	0,00	880,74	455,00	0,00	23,15
	391,00	0,00	58,36	391,00	0,00	23,95
86 (100%)	412,00	0,00	113,33	412,00	0,00	25,83
	523,00	0,00	561,90	523,00	0,00	30,05
	554,00	0,00	105,76	554,00	0,00	27,43
	579,00	0,00	508,46	579,00	0,00	29,90
	541,00	0,00	180,81	540,00	0,18	33,50
95 (110%)	458,00	0,00	880,72	458,00	0,00	32,16
	582,00	0,00	182,86	582,00	0,00	31,81
	376,00	0,00	343,08	376,00	0,00	33,65
	552,00	0,00	410,46	552,00	0,00	41,18
	377,00	0,00	533,85	377,00	0,00	35,73
103 (120%)	477,00	0,00	768,05	477,00	0,00	44,71
	486,00	0,00	462,99	486,00	0,00	36,45
	441,00	0,00	746,83	440,00	0,23	37,04
	474,00	0,42	10923,10	473,00	0,21	44,19
	526,00	0,00	169,96	526,00	0,00	46,61
112 (130%)	475,00	0,00	2702,01	474,00	0,21	99,09
	509,00	0,00	385,85	509,00	0,00	98,15
	490,00	0,00	249,53	490,00	0,00	43,51
	621,00	0,00	352,53	621,00	0,00	102,22
	623,00	0,00	1512,01	623,00	0,00	54,37
120 (140%)	476,00	0,00	489,06	476,00	0,00	55,39
	608,00	0,00	685,16	608,00	0,00	53,62
	499,00	0,00	303,25	499,00	0,00	65,26
	575,00	0,00	7490,70	575,00	0,00	79,50
	539,00	0,00	414,11	538,00	0,19	65,44

Continua na próxima página

Tamanho	BCP			HILS		
	UB	Gap(%)	T(s)	Sol.	Gap_{UB}(%)	T(s)
129 (150%)	521,00	0,00	4721,48	521,00	0,00	69,41
	649,00	0,00	379,84	649,00	0,00	67,70
	528,00	0,00	3447,97	528,00	0,00	72,51
138 (160%)	570,00	0,00	211,96	568,00	0,35	88,66
	612,00	0,00	2397,70	612,00	0,00	88,93
	569,00	0,18	11718,90	569,00	0,00	96,63
146 (170%)	474,00	0,00	1380,11	473,00	0,21	98,12
	541,00	0,00	561,35	540,00	0,18	91,84
	568,00	0,00	3805,30	568,00	0,00	119,68
155 (180%)	582,00	0,00	5886,38	582,00	0,00	107,19
	605,00	0,00	1784,89	605,00	0,00	114,22
	493,00	0,00	1255,79	493,00	0,00	130,01
163 (190%)	630,00	0,00	478,93	629,00	0,16	114,97
	720,00	0,14	10832,80	720,00	0,00	157,72
	525,00	0,00	2031,91	525,00	0,00	140,93
172 (200%)	687,00	0,00	1426,52	687,00	0,00	122,80
	651,00	0,00	6530,42	651,00	0,00	149,90
	651,00	0,00	10080,80	651,00	0,00	141,03
183 (210%)	722,00	0,00	1767,14	722,00	0,00	160,77
	684,00	0,15	10999,40	683,00	0,15	186,48
	637,00	0,00	5304,02	637,00	0,00	146,95
192 (220%)	526,00	0,00	809,38	524,00	0,38	157,44
	572,00	0,00	3772,24	571,00	0,17	161,92
	605,00	0,00	6070,99	605,00	0,00	210,71
201 (230%)	667,00	0,00	5763,56	667,00	0,00	219,48
	553,00	0,00	4485,39	553,00	0,00	211,26
	681,00	0,00	4077,81	681,00	0,00	203,99
210 (240%)	782,00	0,13	12305,50	782,00	0,00	245,17

Continua na próxima página

Tabela 5.11: Resultados computacionais para as instâncias baseadas na XIX EBL

Tamanho	BCP			HILS		
	UB	Gap(%)	T(s)	Sol.	Gap _{UB} (%)	T(s)
28 (30%)	312,00	0,00	1,44	312,00	0,00	2,28
	330,00	0,00	0,27	330,00	0,00	2,08
	364,00	0,00	5,60	364,00	0,00	2,00
	553,00	0,00	0,24	553,00	0,00	2,46
	467,00	0,00	0,38	467,00	0,00	3,05
37 (40%)	339,00	0,00	1,74	339,00	0,00	3,66
	373,00	0,00	21,63	373,00	0,00	3,34
	499,00	0,00	67,80	499,00	0,00	30,65
	362,00	0,00	13,88	363,00	-0,28	3,30
	123,00	0,00	1,66	123,00	0,00	4,91
47 (50%)	463,00	0,00	2270,45	463,00	0,00	5,36
	608,00	0,00	61,21	608,00	0,00	6,62
	781,00	0,00	23,52	781,00	0,00	4,81
	689,00	0,00	13,82	689,00	0,00	5,08
	671,00	0,00	114,59	671,00	0,00	5,29
56 (60%)	818,00	0,00	2325,25	818,00	0,00	9,00
	1562,00	0,00	142,23	1562,00	0,00	7,65
	1139,00	0,00	2,59	1139,00	0,00	8,62
	1118,00	0,00	907,94	1118,00	0,00	9,21
	1067,00	0,00	28,76	1067,00	0,00	7,22
65 (70%)	497,00	0,00	61,05	497,00	0,00	10,74
	551,00	0,00	66,28	551,00	0,00	9,34
	1150,00	0,00	3,88	1150,00	0,00	14,37
	844,00	0,00	60,66	844,00	0,00	14,24
	944,00	0,00	954,65	944,00	0,00	125,01
74 (80%)	1009,00	0,70	10866,30	1002,00	0,69	313,45
	1597,00	0,00	33,29	1597,00	0,00	12,50
	1129,00	0,00	1260,34	1129,00	0,00	14,85
	814,00	0,00	64,57	814,00	0,00	13,37
	1255,00	0,00	39,44	1255,00	0,00	13,05
84 (90%)	653,00	0,00	176,09	650,00	0,46	24,33
	1487,00	0,00	7795,15	1487,00	0,00	18,59
	644,00	0,00	3430,27	644,00	0,00	16,06

Continua na próxima página

Tamanho	BCP			HILS		
	UB	Gap(%)	T(s)	Sol.	Gap _{UB} (%)	T(s)
	1002,00	0,00	1024,64	1002,00	0,00	15,87
	1382,00	0,00	2216,75	1382,00	0,00	16,92
	1270,00	0,24	10849,90	1267,00	0,24	23,81
	1081,00	0,65	10962,20	1073,00	0,74	20,92
93 (100%)	605,00	0,67	10806,80	602,00	0,50	318,53
	767,00	0,79	11039,00	761,00	0,78	35,29
	666,00	0,00	1422,32	659,00	1,05	20,67
	821,00	0,37	10843,80	819,00	0,24	96,40
	1076,00	0,47	10894,00	1072,00	0,37	32,76
102 (110%)	847,00	0,24	10803,60	846,00	0,12	32,48
	1018,00	0,99	10849,10	1009,00	0,88	334,63
	982,00	0,10	10806,40	982,00	0,00	94,40
	1082,00	0,28	10829,90	1080,00	0,18	37,55
	989,00	0,51	10850,20	985,00	0,40	99,06
112 (120%)	1455,00	0,41	10862,10	1450,00	0,34	40,18
	956,00	0,00	504,95	956,00	0,00	36,77
	654,00	0,62	11016,60	650,00	0,61	207,38
	858,00	0,23	10900,00	848,00	1,17	50,05
	1135,00	0,35	10857,30	1132,00	0,26	51,17
121 (130%)	964,00	0,10	10979,00	964,00	0,00	59,69
	1064,00	0,00	7032,84	1064,00	0,00	56,86
	1073,00	0,09	11427,70	1073,00	0,00	58,52
	1169,00	0,17	11059,00	1168,00	0,09	57,53
	782,00	0,13	11421,70	782,00	0,00	61,60
130 (140%)	1641,00	0,98	10880,60	1625,80	0,93	367,08
	934,00	0,00	4142,21	934,00	0,00	68,55
	1352,00	0,60	11239,20	1345,00	0,52	56,85
	923,00	0,65	10973,70	918,00	0,54	124,79
	1015,00	0,50	13076,50	1008,00	0,69	266,19
140 (150%)	1522,00	0,73	11531,80	1511,00	0,72	393,10
	1279,00	0,79	11924,10	1250,40	2,24	396,07
	982,00	0,10	10804,80	982,00	0,00	93,12
	987,00	0,92	11791,20	978,60	0,85	424,63
	1430,00	0,00	7621,68	1430,00	0,00	98,97

Continua na próxima página

Tamanho	BCP			HILS		
	UB	Gap(%)	T(s)	Sol.	Gap _{UB} (%)	T(s)
149 (160%)	960,00	0,21	10934,90	955,20	0,50	402,94
	1850,00	0,11	10943,40	1848,00	0,11	97,25
	906,00	0,55	11015,90	902,00	0,44	425,69
158 (170%)	1128,00	0,18	11705,00	1127,00	0,09	425,81
	1486,00	0,68	10825,60	1464,00	1,48	130,27
	849,00	1,19	13739,30	840,00	1,06	123,06
167 (180%)	1044,00	0,48	11456,60	1040,00	0,38	123,95
	986,00	0,10	12827,50	985,00	0,10	115,37
	1269,00	0,08	12103,80	1265,00	0,32	360,10
177 (190%)	1044,00	0,10	11902,50	1044,00	0,00	168,71
	1461,00	0,07	13035,00	1461,00	0,00	155,67
	929,00	0,11	18132,10	923,00	0,65	138,34
186 (200%)	995,00	0,51	11605,20	986,00	0,90	436,83
	1068,00	0,00	5200,64	1067,00	0,09	199,63
	1348,00	0,97	12189,60	1336,00	0,89	186,51
177 (190%)	1121,00	0,36	17764,80	1112,00	0,80	509,39
	1402,00	0,14	11671,30	1401,00	0,07	472,30
	988,00	0,00	8991,95	986,70	0,13	455,67
186 (200%)	888,00	2,42	14625,10	868,00	2,25	196,91
	991,00	0,10	13645,00	991,00	0,00	342,33
	871,00	3,94	11768,70	831,80	4,50	495,77
186 (200%)	1256,00	0,08	17937,30	1256,00	0,00	263,44
	1045,00	0,10	25030,80	1038,00	0,67	475,33

Tabela 5.12: Resultado computacionais para as instâncias baseadas no LI SBPO

Tamanho	BCP			HILS		
	UB	Gap(%)	T(s)	Sol.	Gap _{UB} (%)	T(s)
29 (10%)	262,00	0,00	0,71	262,00	0,00	2,29
	181,00	0,00	4,42	181,00	0,00	2,09
	208,00	0,00	2,60	208,00	0,00	1,78
	226,00	0,00	2,34	226,00	0,00	1,80
	217,00	0,00	0,91	217,00	0,00	2,05

Continua na próxima página

Tamanho	BCP			HILS		
	UB	Gap(%)	T(s)	Sol.	Gap _{UB} (%)	T(s)
58 (20%)	560,00	6,87	1187,44	524,00	6,43	16,10
	545,00	7,71	958,21	506,00	7,16	6,36
	560,00	10,67	928,18	506,00	9,64	222,36
	560,00	8,74	892,82	515,00	8,04	306,75
	550,00	10,66	838,73	497,00	9,64	15,25
86 (30%)	718,00	10,12	1485,60	652,00	9,19	11,24
	742,00	7,85	1774,45	688,00	7,28	12,12
	767,00	8,64	2002,15	706,00	7,95	12,53
	778,00	11,62	2300,23	697,00	10,41	11,52
	766,00	14,33	2031,13	670,00	12,53	11,19
115 (40%)	956,00	8,88	3076,91	878,00	8,16	151,67
	914,00	6,28	3405,93	860,00	5,91	29,47
	923,00	8,46	3079,64	851,00	7,80	24,58
	975,00	8,82	3545,78	896,00	8,10	27,89
	896,00	0,00	1421,93	896,00	0,00	86,00
144 (50%)	1017,00	7,62	3177,69	945,00	7,08	52,42
	1038,00	8,81	2824,94	954,00	8,09	58,74
	993,00	10,33	3201,58	900,00	9,37	56,59
	1003,00	8,20	3034,54	927,00	7,58	54,66
	1030,00	7,97	2906,51	954,00	7,38	42,24
173 (60%)	955,00	0,00	330,22	955,00	0,00	82,64
	1036,00	0,00	1099,57	1036,00	0,00	84,92
	955,00	0,00	4040,26	955,00	0,00	73,88
	1036,00	7,47	5242,25	964,00	6,95	93,42
	1117,00	0,00	1224,75	1117,00	0,00	84,06
202 (70%)	1145,00	7,61	9656,19	1064,00	7,07	181,95
	1208,00	6,34	7777,35	1136,00	5,96	175,73
	1266,00	6,39	7605,59	1190,00	6,00	146,05
	1064,00	6,29	10758,90	1001,00	5,92	166,99
	1091,00	0,00	6641,14	1091,00	0,00	211,09
230 (80%)	1309,00	8,18	9043,25	1210,00	7,56	280,43
	1156,00	0,00	5069,30	1156,00	0,00	243,14
	1183,00	0,00	2354,49	1183,00	0,00	324,43
	1111,00	0,00	1288,68	1111,00	0,00	293,56

Continua na próxima página

Tamanho	BCP			HILS		
	UB	Gap(%)	T(s)	Sol.	Gap _{UB} (%)	T(s)
	1165,00	0,00	489,27	1165,00	0,00	287,22
	1175,00	0,00	4825,60	1175,00	0,00	417,46
	1337,00	0,00	1229,45	1337,00	0,00	475,23
259 (90%)	1229,00	0,00	5265,43	1229,00	0,00	414,06
	1283,00	0,00	175,06	1283,00	0,00	369,19
	1229,00	0,00	6466,61	1229,00	0,00	386,20
	1583,00	1,54	10850,70	1560,00	1,45	603,60
	1443,00	0,07	20255,30	1443,00	0,00	607,58
288 (100%)	1407,00	2,03	24378,20	1380,00	1,92	941,11
	1533,00	0,07	12490,50	1533,00	0,00	602,16
	1515,00	0,60	16166,30	1506,00	0,59	621,85

Tabela 5.13: Resultado computacionais para as instâncias reais

Evento/Tamanho	BCP			HILS		
	UB	Gap(%)	T(s)	Sol.	Gap _{UB} (%)	T(s)
LARS/86	1194,00	0,00	57,68	1194,00	0,00	20,90
EBL/93	1244	0,72	10804,10	1236,60	0,59	31,65
SBPO/288	2800,00	4,40	15405,90	2683,00	4,18	666,80

5.2.2 Resultados obtidos em eventos reais

Em um primeiro momento, utilizou-se eventos reais que já haviam acontecido para que fosse possível comparar os resultados obtidos através da programação gerada manualmente com a programação gerado pelo HILS. Utilizou-se como instâncias as edições 2016 e 2017 do SBPO.

Os dados foram obtidos através dos *sites* dos eventos e de planinhas com a listagem dos trabalhos cedidas por um dos organizadores. O SBPO de 2016 teve um total de 288 trabalhos, 22 temas e 93 sessões. Já o SBPO 2017 é uma instância de maior porte, tendo contado com 324 trabalhos, 22 temas e 95 sessões. Nesses dois eventos, a programação das sessões técnicas foi realizada de forma manual. Para comparação com a solução proposta, a função objetivo foi calculada para as duas edições do evento. As funções objetivo da solução manual das edições de 2016 e 2017 foram 718 e 1926,

respectivamente. O algoritmo híbrido, por sua vez, atingiu os valores de 3048 e 5495, respectivamente. Para a programação dessas duas edições foi levando em consideração todos os autores do trabalho durante.

5.2.3 Aplicações

Ao longo da pesquisa, surgiram oportunidades de gerar a programação de alguns eventos. A seguir são apresentados os eventos em que o HILS serviu como ferramenta para ajudar na geração da programação.

A primeira aplicação do HILS em um evento foi o L SBPO (2018). Esta edição contou com 283 trabalhos, 22 temas e 85 sessões. A heurística além de gerar a programação também rotulou os temas de cada sessão. Na programação gerada, apenas uma sessão apresentou um trabalho fora do tema dos demais. Foi disponibilizada uma interface *web* (Figuras A.1–A.3) para que os organizadores do evento pudessem editar e baixar a programação gerada. A programação final foi mantida quase que por completa com exceção de alguns poucos trabalhos que foram remanejados por questões de disponibilidade de alguns apresentadores.

Outro evento foi o XV LARS (2018), que contou com 86 trabalhos, 31 temas e 18 sessões. Todas as sessões foram contempladas com trabalhos relacionados. Para esse evento, foi disponibilizada a mesma interface *web* que foi utilizada pela organização do SBPO 2018. Porém, os organizadores mantiveram a programação gerada pelo HILS.

Foi realizada a programação da EBL (2019). Esta edição contou com 93 trabalhos, 8 temas e 21 sessões. Para esta edição do evento, criou-se uma nova interface *web* (Figuras A.4–A.6) que permitia aos organizadores juntarem pares de trabalhos que deveriam aparecer juntos em uma sessão. Para este evento, o algoritmo foi atualizado para levar em consideração a disponibilidade dos autores. Isso foi feito alterando a função objetivo do checador de viabilidade (4.1) para que minimizasse as preferências dos apresentadores por dia e horário. Um total de 19 restrições desse tipo foram utilizadas e apenas para um autor não foi possível atender todas as suas restrições. A programação final sofreu algumas alterações, pois alguns apresentadores tinham compromissos em mesas redondas e outras atividades do evento que não foram consideradas pelo algoritmo.

Mais uma vez, em 2019, a programação do LI SBPO foi realizada utilizando o

HILS. Essa edição do evento contou com 288 trabalhos, 26 temas e 91 sessões. A interface *web* foi disponibilizada novamente para permitir a edição da programação. A programação foi mantida quase que integralmente, com exceção de alguns trabalhos que foram remanejados para sessões especiais.

Capítulo 6

Conclusão

Neste trabalho foram propostos duas formulações matemáticas e um algoritmo híbrido para o PPSTC. Para testar a efetividade dos modelos e do algoritmo híbrido, diversos experimentos foram realizados. Criou-se um gerador de instâncias, a partir do qual criou-se 230 instâncias artificiais. As instâncias foram geradas com base em 3 eventos (LARS, EBL e SBPO).

As formulações não escalaram bem, tornando seu uso para casos reais inviável. O algoritmo híbrido, por sua vez, apresentou um desempenho bem superior. Tal desempenho permitiu que fosse possível aplicá-lo na realização de alguns eventos, gerando soluções que foram utilizadas na prática, com pouca ou nenhuma alteração. Para as instâncias geradas em que o ótimo é conhecido, o HILS conseguiu atingir 83.67% deles.

As aplicações realizadas em eventos reais demonstram que o HILS pode ser utilizado na prática diminuindo consideravelmente o tempo para programar as sessões técnicas se comparado com a realização manual. As soluções geradas para tais eventos sofreram pouca ou nenhuma alteração, o que mostra que a solução gerada é satisfatória para o uso real.

Algumas sugestões para trabalhos futuros incluem melhorar o limite de temas por faixa de horário e dia, que poderiam ter uma forma de ajustá-los melhor de acordo com características da instância, visto que dependendo dos limites adotados, é possível levar a inviabilidades. Para tanto, uma alternativa poderia ser tentar minimizar o número

de trabalhos paralelos com o mesmo número de temas ao invés de utilizar limites.

Outra sugestão é adicionar novas restrições para que o algoritmo possa atender conferências com necessidades não contempladas na versão apresentada como, por exemplo, poderia ser incluída a programação do coordenador das sessões. Com a adição de novas restrições, a opção de poder selecionar quais seriam utilizadas tornaria o algoritmo mais flexível e aplicável a mais eventos, visto que nem todos possuem as mesmas características.

Referências Bibliográficas

- BENTLEY, J. J. Fast algorithms for geometric traveling salesman problems. *ORSA Journal on Computing*, v. 4, n. 4, p. 387–411. doi: 10.1287/ijoc.4.4.387, 1992.
- BULHÕES, T.; CORREIA, R.; SUBRAMANIAN, A. Conference scheduling: a clustering-based approach. Working paper, 2019.
- BULHÕES, T.; DE SOUSA FILHO, G. F.; SUBRAMANIAN, A.; DOS ANJOS F. CABRAL, L. Branch-and-cut approaches for p-cluster editing. *Discrete Applied Mathematics*, v. 219, p. 51 – 64. ISSN 0166-218X, 2017.
- EDIS, E.; SANCAR EDIS, R. An integer programming model for the conference timetabling problem. *Celal Bayar University Journal of Science*, v. 9, p. 55–62, 2013.
- EGLESE, R. W.; RAND, G. K. Conference seminar timetabling. *Journal of the Operational Research Society*, v. 38, n. 7, p. 591–598, 1987.
- GRÖTSCHEL, M.; WAKABAYASHI, Y. Facets of the clique partitioning polytope. *Mathematical Programming*, v. 47, n. 1, p. 367–387, 1990.
- IBRAHIM, H.; RAMLI, R.; HASNAN HASSAN, M. Combinatorial design for a conference: Constructing a balanced three-parallel session schedule. *Journal of Discrete Mathematical Sciences and Cryptography*, v. 11, p. 305–317, 2008.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 671–680. ISSN 00368075, 1983.

- LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. *Handbook of Metaheuristics*, Chapter Iterated Local Search: Framework and Applications, p. 363–397. Springer, 2010.
- MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. *Computers & Operations Research*, v. 24, n. 11, p. 1097–1100. ISSN 0305-0548, 1997.
- MORI, Y.; TANAKA, M. A hybrid grouping genetic algorithm for timetabling of conference programs. *Proceedings of the Annual Conference of the Institute of Systems, Control and Information Engineers*, v. SCI02, p. 223–223, 2002.
- NICHOLLS, M. G. A small-to-medium-sized conference scheduling heuristic incorporating presenter and limited attendee preferences. *Journal of the Operational Research Society*, v. 58, n. 3, p. 301–308, 2007.
- POTTHOFF, R. F.; BRAMS, S. J. Scheduling of panels by integer programming: Results for the 2005 and 2006 new orleans meetings. *Public Choice*, v. 131, n. 3, p. 465–468, 2007.
- POTTHOFF, R. F.; MUNGER, M. C. Use of integer programming to optimize the scheduling of panels at annual meetings of the public choice society. *Public Choice*, v. 117, n. 1, p. 163–175, 2003.
- QUESNELLE, J.; STEFFY, D. Scheduling a conference to minimize attendee preference conflicts. HANZALEK, Z.; KENDALL, G.; MCCOLLUM, B.; SUCHA, P. (Eds.), *In proceedings of the 7th Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2015), 25 - 28 Aug 2015, Prague, Czech Republic*, p. 379–392. Paper, 2015.
- SAMPSON, S. E. Practical implications of preference-based conference scheduling. *Production and Operations Management*, v. 13, n. 3, p. 205–215, 2004.
- SAMPSON, S. E.; WEISS, E. N. Increasing service levels in conference and educational scheduling: A heuristic approach. *Management Science*, v. 41, n. 11, p. 1816–1825, 1995.

TANAKA, M.; MORI, Y.; BARGIELA, A. Granulation of keywords into sessions for timetabling conferences. *Soft Computing and Intelligent Systems SCIS'2002, Tsukuba, Japan, October 2002*, 2002.

THOMPSON, G. M. Improving conferences through session scheduling preferences. *Cornell Hotel and Restaurant Administration Quarterly*, v. 43, p. 71–76, 2002.

VANGERVEN, B.; FICKER, A. M.; GOOSSENS, D. R.; PASSCHYN, W.; SPIEKEMA, F. C.; WOEGINGER, G. J. Conference scheduling — a personalized approach. *Omega*, v. 81, p. 38 – 47, 2018.

Apêndice A

Images das aplicações web

Este apêndice apresenta as imagens tiradas da aplicação desenvolvidas para o SBPO e EBL.

The screenshot shows the SBPO 2018 event program interface. At the top, there is a navigation bar with the text "SBPO 2018", "Programação", "Temas por Sessão", and "Choque de Autores". On the right side of the navigation bar, there is a user email "puca@email.com" and a "Log off" link. Below the navigation bar, the main heading is "Programação do Evento SBPO 2018". Underneath this heading, there is a blue button labeled "Exportar Programação". The main content area is titled "terça-feira - 07/08/2018 09:30 às 10:30". It contains two session blocks. The first session is "Sessão 1: IND-1 / terça-feira - 07/08/2018 09:30 às 10:30 / Capacidade: 3". It lists three topics: 84129 "AVALIAÇÃO DE MODELOS DE PREVISÃO DE DEMANDA PARA PEÇAS DE REPOSIÇÃO NA INDÚSTRIA AUTOMOTIVA (IND – PO na Indústria, L&T – Logística e Transportes)" by Ana Pantaleão, Nadjib BRAHIMI; 84781 "BALANCEAMENTO DE UMA LINHA DE MANUFATURA ROBOTIZADA COM CARACTERÍSTICAS SIMÉTRICAS POR MEIO DE PROGRAMAÇÃO MATEMÁTICA (IND – PO na Indústria)" by Rafael Gobbi Molina, Thiago Cantos Lopes, Luiz Carlos de Abreu Rodrigues, Leandro Magatão; and 84730 "Uma Metaheurística GRASP para o Planejamento de Colheitas e de Chegadas de Caminhões a Usinas de Processamento Agroindustrial (AG&MA – PO na Agricultura e Meio Ambiente, IND – PO na Indústria, L&T – Logística e Transportes)" by Marcela González-Araya. Each topic entry has a blue "Trocar" button and a green "Mover" button. The second session is "Sessão 2: MH-1 / terça-feira - 07/08/2018 09:30 às 10:30 / Capacidade: 3", listing topic 84991 "Algoritmo genético de chaves aleatórias segundo distribuição de Levy para otimização global (MH – Metaheurísticas)" with "Trocar" and "Mover" buttons.

Figura A.1: Tela com exibição da solução com possibilidade de edição

Tópicos por Sessão - SBPO 2018

Horários	Programação										
terça-feira - 07/08/2018 09:30 às 10:30	IND-1 (1)	MH-1 (2)	EN-1 (3)	TAG-1 (4)	OC-1 (5)	SIM-1 (6)	EN-2 (7)	ADM-1 (8)	PM-1 (9)	ADM-2 (10)	SA-1 (11)
terça-feira - 07/08/2018 15:00 às 16:00	L&T-1 (12)	OC-2 (13)	OC-3 (14)	ADM-3 (15)	MH-2 (16)	AG&MA-1 (17)	AD&GP-1 (18)	SIM-2 (19)	TEL&SI-1 (20)	GF-1 (21)	SA-2 (22)
terça-feira - 07/08/2018 16:30 às 18:10	ADM-4 (23)	TAG-2 (24)	OA-1 (25)	L&T-2 (26)	IND-2 (27)	OC-4 (28)	GF-2 (29)	MP-1 (30)	EN-3 (31)	SIM-3 (32)	EN-4 (33)
quarta-feira - 08/08/2018 09:30 às 10:30	AdP-1 (34)	IND-3 (35)	EDU-1 (36)	TAG-3 (37)	L&T-3 (38)	ADM-5 (39)	MH-3 (40)	OC-5 (41)	EST-1 (42)	EST-2 (43)	OC-6 (44)
quarta-feira - 08/08/2018 15:00 às 16:00	DEA-1 (45)	OC-7 (46)	P&G-1 (47)	MH-4 (48)	IND-4 (49)	OA-2 (50)	EN-5 (51)	ADM-6 (52)	EN-6 (53)	GF-3 (54)	
quarta-feira - 08/08/2018 16:30 às 18:10	TAG-4 (55)	PM-2 (56)	OC-8 (57)	EN-7 (58)	DEA-2 (59)	MH-5 (60)	OC-9 (61)	L&T-4 (62)	P&G-2 (63)		
quinta-feira - 09/08/2018 09:30 às 10:30	EN-8 (64)	OC-10 (65)	ADM-7 (66)	OC-11 (67)	L&T-5 (68)	OA-3 (69)	MH-6 (70)	SIM-4 (71)	DEA-3 (72)	EN-9 (73)	SA-3 (74)
quinta-feira - 09/08/2018 11:00 às 12:00	ADM-8 (75)	EN-10 (76)	MH-7 (77)	TAG-5 (78)	L&T-6 (79)	ADM-9 (80)	TEL&SI-2 (81)	AD&GP-2 (82)	MH-8 (83)	EST-3 (84)	L&T-7 (85)

Programação SBPO

Figura A.2: Exibição dos temas das sessões por faixa de horário

quarta-feira - 08/08/2018 09:30 às 10:30 (11)

Sessão: 34 - AdP - PO na Administração Pública

Sessão: 35 - IND - PO na Indústria

Sessão: 36 - EDU - PO na Educação

Sessão: 37 - TAG - Teoria e Algoritmos em Grafos

Autores com mais de um trabalho nesta sessão:

Uéverton Souza (2)

Sessão: 38 - L&T - Logística e Transportes

Sessão: 39 - ADM - Apoio à Decisão Multicritério

Sessão: 40 - MH - Metaheurísticas

Sessão: 41 - OC - Otimização Combinatória

Sessão: 42 - EST - Estatística

Sessão: 43 - EST - Estatística

Sessão: 44 - OC - Otimização Combinatória

Autores com mais de um trabalho nesta faixa de horário:

Nenhum

quarta-feira - 08/08/2018 15:00 às 16:00 (10)

Sessão: 45 - DEA - Análise Envoltória de Dados

Sessão: 46 - OC - Otimização Combinatória

Sessão: 47 - P&G - PO na Área de Petróleo e Gás

Sessão: 48 - MH - Metaheurísticas

Sessão: 49 - IND - PO na Indústria

Sessão: 50 - OA - Outras aplicações em PO

Sessão: 51 - EN - PO na Área de Energia

Figura A.3: Exibição de todas as sessões de uma faixa de horário e choques de autores, caso exista

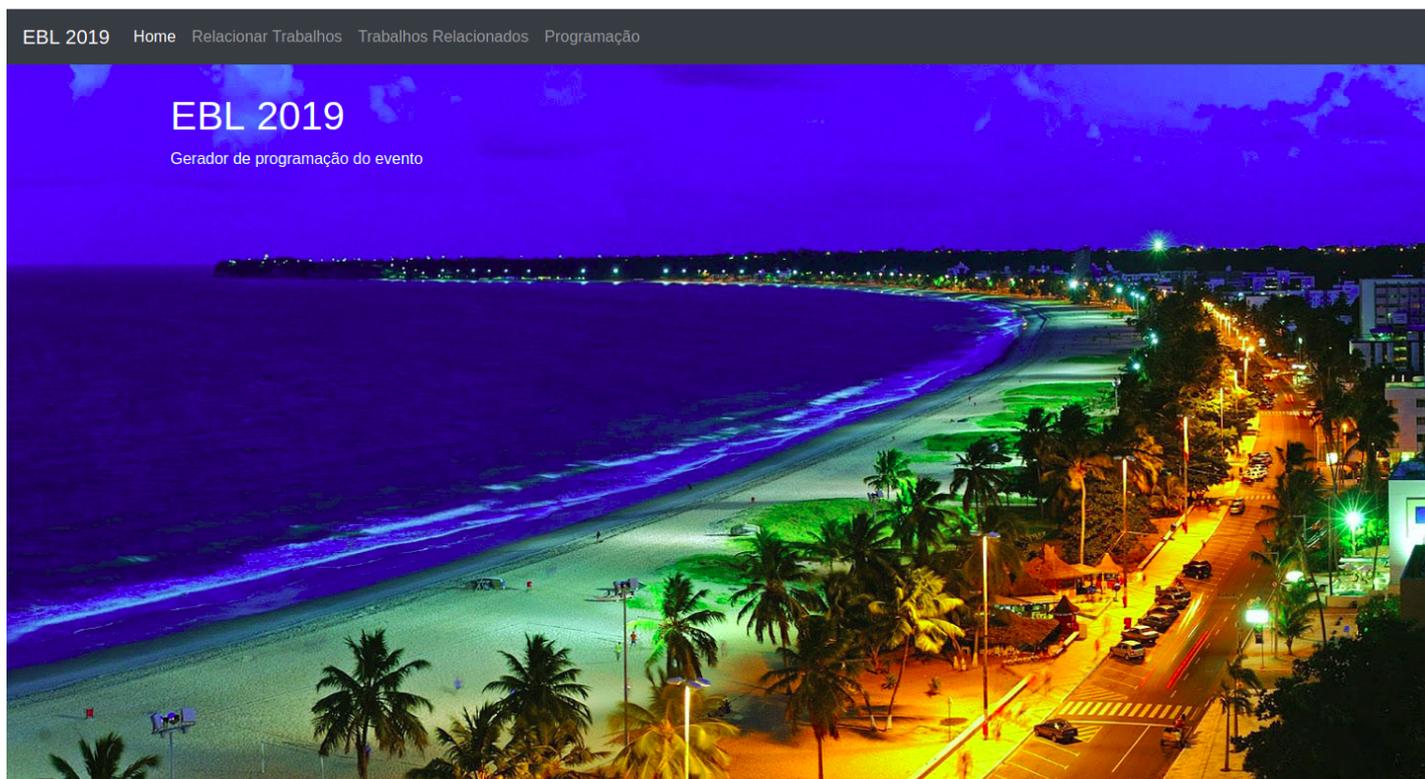
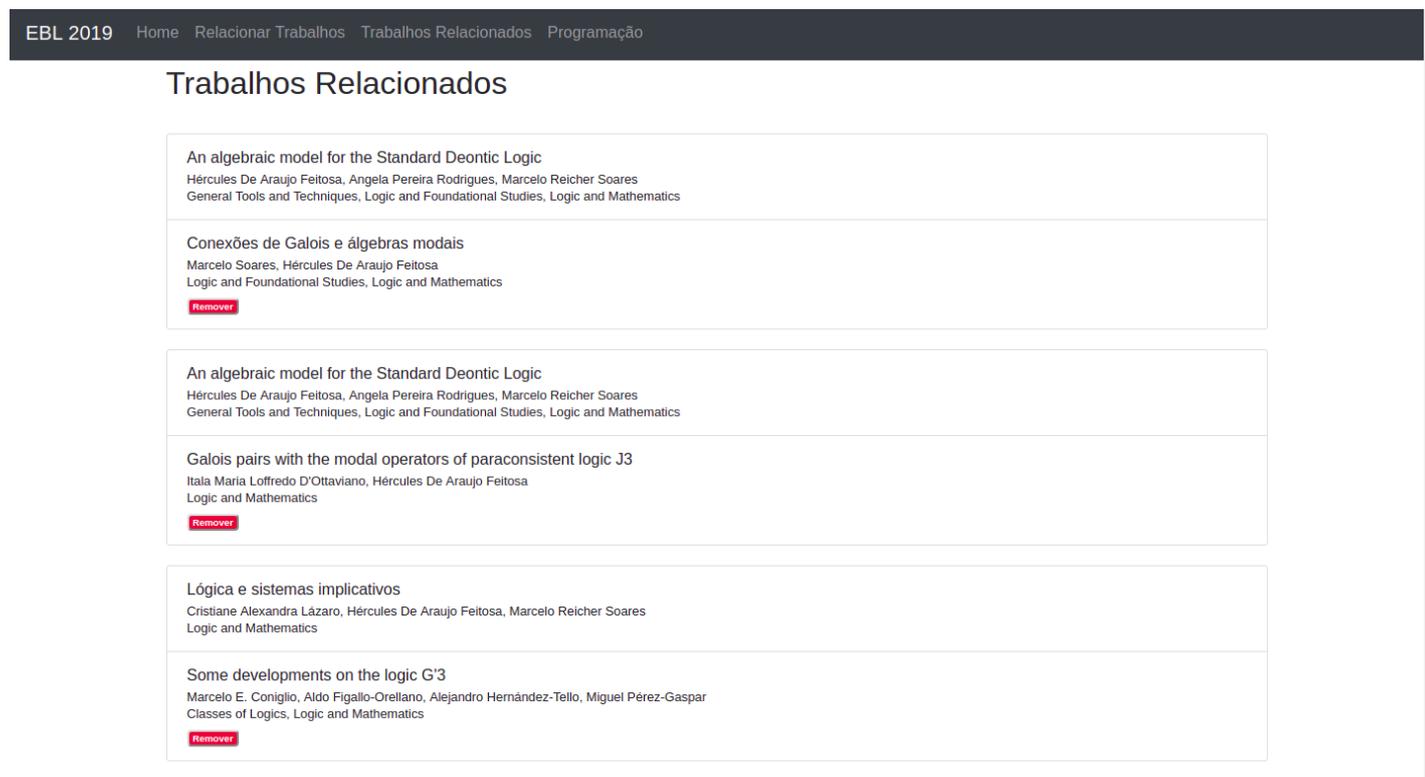


Figura A.4: Tela inicial da aplicação do evento EBL 2019

#	Título	Autores	Temas
<input checked="" type="checkbox"/> 1	An algebraic model for the Standard Deontic Logic	Hércules De Araujo Feitosa, Angela Pereira Rodrigues, Marcelo Reicher Soares	General Tools and Techniques, Logic and Foundational Studies, Logic and Mathematics
<input checked="" type="checkbox"/> 2	Lógica e sistemas implicativos	Cristiane Alexandra Lázaro, Hércules De Araujo Feitosa, Marcelo Reicher Soares	Logic and Mathematics
<input checked="" type="checkbox"/> 3	Conexões de Galois e álgebras modais	Marcelo Soares, Hércules De Araujo Feitosa	Logic and Foundational Studies, Logic and Mathematics
<input type="checkbox"/> 4	Teaching Logic Historically	Danielle Macbeth	History and Philosophy of Logic, Logic and Mathematics, Philosophy and Philosophical Logic
<input type="checkbox"/> 5	Compiling Reo Certified Code	Erick Grilo, Bruno Lopes	General Tools and Techniques, Logic in Computer Science
<input type="checkbox"/> 6	Verificação de modelos Reo com nuXmv	Daniel Arena Toledo, Bruno Lopes	General Tools and Techniques, Logic in Computer Science
<input type="checkbox"/> 7	Compiling C code to PDL	Philippe Araujo, Allan Santana, Bruno Lopes	Logic in Computer Science
<input type="checkbox"/> 8	Teaching Fragments of Natural Deduction	Felipe S. Albarelli, Rodolfo C. Ertola Biraben	Classes of Logics
<input type="checkbox"/> 9	Semânticas não-determinísticas para lógicas não-clássicas: uma abordagem da perspectiva de Teoria de Modelos e de Álgebra Universal	Guilherme Toledo, Marcelo Coniglio	Classes of Logics, Logic and Mathematics

Figura A.5: Tela que permite selecionar trabalhos que devem aparecer juntos



The screenshot shows a web interface with a dark header bar containing the text "EBL 2019 Home Relacionar Trabalhos Trabalhos Relacionados Programação". Below the header, the title "Trabalhos Relacionados" is displayed. The main content area consists of five vertically stacked white boxes, each representing a related work. Each box contains the title of the work, the authors' names, and the journal or publication name. A small red button labeled "Remover" is located at the bottom left of each box.

Work Title	Authors	Journal/Publication	Action
An algebraic model for the Standard Deontic Logic	Hércules De Araujo Feitosa, Angela Pereira Rodrigues, Marcelo Reicher Soares	General Tools and Techniques, Logic and Foundational Studies, Logic and Mathematics	Remover
Conexões de Galois e álgebras modais	Marcelo Soares, Hércules De Araujo Feitosa	Logic and Foundational Studies, Logic and Mathematics	Remover
An algebraic model for the Standard Deontic Logic	Hércules De Araujo Feitosa, Angela Pereira Rodrigues, Marcelo Reicher Soares	General Tools and Techniques, Logic and Foundational Studies, Logic and Mathematics	Remover
Galois pairs with the modal operators of paraconsistent logic J3	Itala Maria Loffredo D'Ottaviano, Hércules De Araujo Feitosa	Logic and Mathematics	Remover
Lógica e sistemas implicativos	Cristiane Alexandra Lázaro, Hércules De Araujo Feitosa, Marcelo Reicher Soares	Logic and Mathematics	Remover
Some developments on the logic G'3	Marcelo E. Coniglio, Aldo Figallo-Orellano, Alejandro Hernández-Tello, Miguel Pérez-Gaspar	Classes of Logics, Logic and Mathematics	Remover

Figura A.6: Exibição das preferências dos trabalhos que devem aparecer juntos